DOCUMENT RESUME

ED 053 531 EM 009 048

AUTHOR Homeyer, Fred C.
TITLE Development and Evaluation of an Automated Assembly Language Teacher.
INSTITUTION Texas Univ., Austin. Computer-Assisted Instruction Lab.
SPONS AGENCY National Science Foundation, Washington, D.C.
REPORT NO TR-3
PUB DATE May 70
NOTE 240p.

EDRS PRICE EDRS Price MF-$0.65 HC-$9.87
DESCRIPTORS Comparative Analysis, *Computer Assisted Instruction, Computer Programs, *Computer Science Education, Conventional Instruction, *Course Evaluation, Curriculum Development, Formative Evaluation, Input Output Devices, Instructional Design, *Programing, *Programing Languages, Student Attitudes, Summative Evaluation
IDENTIFIERS ELASTIC, *Expandible Language Aiding Student Instruction

ABSTRACT

A programing language called ELASTIC (Expandible Language for Aiding Student Instruction in Computing) has been developed which incorporates many of the basic features and concepts of a typical assembly language. ELASTIC was designed for use in a computer-assisted instruction (CAI) undergraduate course in computer programing and was written in a version of FORTRAN. It has a modular design and is suitable for a batch-process time-shared computer facility. A prototype (CAI) course to teach ELASTIC was developed. A formative evaluation of the course resulted in the addition of facilities for processing student responses, removal or clarification of ambiguous statements, and expansion of curriculum dealing with programing techniques and procedures. The revised instructional program underwent a summative evaluation in which the CAI course was compared with a lecture course and the batch-processing mode was compated with the use of a teletype. Among the results of this evaluation were that CAI groups completed the course instruction twice as fast as the lecture groups and that members of the CAI/teletype group ran a far greater number of jobs through the computer than did students in any other group. Supplementary information about the instructional program and about ELASTIC is appended. (JY)

THE UNIVERSITY OF TEXAS AT AUSTIN
Computer Assisted Instruction Laboratory
AUSTIN

DEVELOPMENT AND EVALUATION OF AN AUTOMATED ASSEMBLY LANGUAGE TEACHER

TECHNICAL REPORT NO. 3

Fred C. Homeyer

May 1970

Supported By:

THE NATIONAL SCIENCE FOUNDATION
Grant GJ 509 X

The University of Texas at Austin
Computer-Assisted Instruction Laboratory
C. Victor Bunderson, Director
Austin, Texas 78712

ACKNOWLEDGEMENTS

I wish to express my sincere appreciation to my supervising professor, Dr. C. Victor Bunderson for his suggestions and assistance throughout this effort. Gratitude is also due to the members of my supervisory committee, Dr. Terrence W. Pratt, Dr. W. W. Bledsoe and Dr. Jack M. Knutson.

Partial support for this work was provided by National Science Foundation Grant GJ-509X, Empirical and Theoretical Foundations of Instructional Design for Computer-Based Instruction Systems, A Program of Research and Development, and by National Science Foundation Grant GU-1598, University Science Development Program.

F. C. H.

Austin, Texas

June, 1970

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

FIGURE PAGE

CHAPTER I

Problem and Approach

Social and Institutional Need

Due to the technology explosion which our country has witnessed in the past decade, a greater emphasis has been placed on the use of machines, and more especially computers, to perform the routine clerical tasks as well as assisting in more complex activities. The tremendous increase in information processing capabilities brought about by the advent of computers has contributed greatly to scientific experimentation which in turn has amplified the rate of technological development.

The communication between man and machine is one of the weaker aspects in the conversion to widespread computer utilization. It has been estimated that by the middle of the next decade there will be a shortage of several hundred thousand skilled programmers to command the computers. It seems evident that the responsibility for satisfying the need for skilled technical personnel rests primarily upon the educational institutions, both public and private.

The fact is well-accepted that a certain degree of aptitude is necessary for programming a computer. It is also believed that the best programmers usually begin their training, with respect to programming languages, by learning a basic assembly language. By mastering a language of this type at first, a certain degree of insight is gained for learning the higher, more sophisticated programming languages. Therefore the need exists for a basic assembly language which is very "typical" and which incorporates the fundamental concepts of programming.

In addition to the need for a suitable assembly language, there is a need for a pedagogical method to effectively train the maximum number of people in a minimum time period and still retain economic feasibility.

Justification

Computer-assisted instruction (CAI) methodology should more than satisfy the need for an effective instructional medium. At first glance this suggestion seems impractical due to the current structure of the educational system in the United States; however, through the use of new and revolutionary teaching techniques, such as CAI, the goal of satisfying the demand for the programmers can be realized. Although the potential of CAI has been expounded upon by many in the last few years, the achievement of this potential can only be accomplished through more widespread applied research and demonstration of the method.

Significant research in computer-assisted instruction is being performed at the University of Texas at Austin. The results of this research as well as the availability of adequate facilities at Texas has prompted the use of CAI in the teaching of assembly language programming.

The need for a basic assembly language as mentioned above has also been partially satisfied : the University of Texas. A programming language called ELASTIC has been developed and is presently being taught in an undergraduate course. ELASTIC incorporates many basic features and concepts of a "typical" assembly language. It has been hypothesized, although never statistically verified, that students who possess a working knowledge of ELASTIC have less difficulty learning additional programming languages than those students never exposed to ELASTIC. ELASTIC provides the student with an insight into computing concepts that is difficult to obtain elsewhere.

ELASTIC (Expandible Language for Aiding Student Instruction in Computing) is a programming language [17] designed, written and modified by T. W. Pratt, C. M. Williams, A. Pearson, E. Pearson, G. Autrey and F. Homeyer. The primary motivation underlying the design of this language was the need for a comprehensive assembly language for use in an undergraduate course in computer programming.

The design of ELASTIC manifests the fact that consideration was given to both pedagogical and practical aspects of assembly language instruction. A number of convenient and advantageous programming features were built into the system. A significant property of ELASTIC is the fact that it is relatively easy to teach and since the system is written in FORTRAN, it can be used with a minimum modification on any computer that has a FORTRAN compiler. A logical consequence of this easy adaptability is that teachers do not have to be retained for each new machine that is acquired by the institution since the ELASTIC system can merely be transferred to the new computer. Included in the ELASTIC language are concepts which can be applied to many other computer languages so that students who have learned ELASTIC possess a foundation which provides a basis for learning other assembly languages with a minimum of effort.

A modular design was adopted for the basic structure of ELASTIC. This modular design permits alteration of the word size, memory capacity and instruction set of the language through minor modifications to the FORTRAN program that is the ELASTIC system. Additional information on ELASTIC may be found in Chapter II and in Appendix D of this paper.

The characteristics and properties of a computer can inherently provide increased justification for its use when the computer functions as an aid in the instructional situation. For example, interfacing instruction in a language with the execution of programs in the language becomes potentially feasible. The use of CAI can contribute the features of generality, compactness and manipulability to an instructional course. In addition, the instruction received by students is consistent from semester to semester when CAI is employed. Through CAI, a student can progress through a course of instruction at his own pace and at times which fit conveniently into his schedule. In other words, the student does not have to "be in the mood" for learning assembly language every Monday, Wednesday and Friday at 10:00 P. M. as a formal lecture situation might dictate because the student would have access to the computer and the CAI course during all regular "production time" that the computer is in operation. Finally and most important, the flexibility of computer-assisted instruction can provide each student with concept development and practice in programming according to his individual needs.

Goals

A number of goals were established relative to the overall instructional experiment upon which this paper reports. These goals will be enumerated below.

1. The student, upon completion of the CAI course, will have adequate programming ability in assembly language.

2. The student will become more highly motivated toward using computers as a result of participation in the instructional experiment.

3. Through the use of the instructional program, the student will develop an appreciation for instructional uses of the computer.

Approach

Following an in-depth examination of the ELASTIC language and related techniques the approach to the problem of teaching assembly language by CAI took the form of a number of logically dependent sequences. The first activity to be performed was an extensive review of the literature as it related to using CAI to teach computer programming. The major results of this review are presented in the next section of this chapter. It was found that in the major efforts in which CAI was used to teach programming, a procedure-oriented language such as FORTRAN or COBOL was taught rather than an assembly language. In addition to this finding, it was discovered that very few of these CAI courses had as yet been extensively evaluated to determine their validity or effectiveness of instruction.

The next activity initiated in this approach consisted of designing, developing and implementing a program to teach the ELASTIC programming language. The instructional program was designed to operate in a time-shared environment using a computer-assisted instruction mode for presentation of curriculum to the student. It was intended that the CAI course provide complete and effective instruction in the concepts and techniques of assembly language programming. This endeavor is discussed in Chapter II of this paper.

The formative evaluation which is described in Chapter III constituted the next major step in the experiment. The purpose of the evaluation was to determine the effectiveness of the instructional program according to the stated objectives established for it. The data collected in this evaluation formed the basis for the major modifications that were made in both the instructional program and the instructional system used to drive the program. These modifications are also discussed in Chapter III.

The final activity was a summative evaluation in which the computer-assisted instruction course and facilities for on-line ELASTIC program execution were compared to a typical classroom lecture type of curriculum presentation. This study and its findings are proffered in Chapter V.

Related Studies

The purpose of this section is to report the findings of the literature review mentioned in the section above. The major difficulty encountered in this review was that much of the work being performed in the area of using CAI to teach programming is still in some state of design or implementation. It is for this reason that very little published material exists. In fact the information obtained on almost all of the projects discussed below was obtained through personal correspondence with various universities and colleges. It should also be understood that the projects to be discussed by no means constitute a review of all efforts in the area; however, they do provide an overview that should illustrate the "flavor" of the majority of such projects.

The only CAI course in programming upon which an evaluation study has been reported, as far as the author has been able to determine, was conducted by J. Schurdak and appeared in the American Educational Research Journal [27] in 1967. The instructional program developed in this study was used to teach the FORTRAN language. The evaluation procedure took the form of comparing three groups of students who received instruction in FORTRAN through one of three instructional treatments. These treatments were computer-assisted instruction, programmed instruction, and a conventional textbook. The CAI course was developed by Schurdak while the programmed instruction text was written by Stephen Plumb and was entitled, "FORTRAN- Self Teaching."

The conventional textbook used in the third treatment mentioned above was the 1961 edition of, "A Guide to FORTRAN Programming," by Daniel D. McCracken.

Upon completion of the course in FORTRAN, the students were given an achievement test prepared by Schurdak and a colleague. From the results of the test it was reported that the students who received instruction through CAI scored much higher than students in the other two groups.

From the data obtained in the experiment, Schurdak made a number of unsupported generalizations and assertions. These generalizations as well aa certain claims and objectives which were set forth, resulted in this report's receiving a great deal of criticism. Kopstein and Seidel [20] discuss many shortcomings of Schurdak's study and in addition provide a number of suggested criteria by which work in CAI can be judged. Among these criteria is the necessity for sepcification of terminal behavorial objectives which can be used to determine the effectiveness of an instructional program. In addition it was stated that it should be realized that the computer is merely a tool to guide the instructional process and does not constitute an instructional process in itself.

The Brighton College of Technology has developed a computer-assisted instruction scheme whereby the student receives instruction in FORTRAN from a programmed instruction (PI) manual and then uses the computer to test the programs that he writes [15]. The computer used is an ICT 1900. The student's grasp of programming is demonstrated by how often and in what manner he uses the computer. Three modes of program execution as well as extensive record keeping facility are provided for every student through the use of the GEORGE operating system.

Program assignments are presented throughout the PI course. These programs are then written by the student and executed in TEST, COMPILE or

JUDGE mode. The COMPILE mode is used to translate the FORTRAN program to object code and give diagnostics concerning syntax errors. The TEST mode is an execution mode in which the student provides his own test data for his program. When he feels that the program is complete and error-free, the student can execute his job in JUDGE mode. In this mode, the GEORGE operating system supplies a set of test data for the program and informs the student as to the success or failure of his job. The operating system updates the student's individual record file everytime he runs a program. Through examination of the student records, the teaching staff can determine the progress and success of all students. This course is being used in a freshman course at Brighton. No data concerning the effectiveness of this course was available at the time of this writing.

Another system used to teach FORTRAN was developed by Aerospace Corporation. This system is called PROCTOR and is in effect a computer program written in FORTRAN and ASCENTF for execution on Control Data Corporation (CDC) 6000 computers [1]. It essentially makes a CDC 211 cathode-ray tube terminal act like a teaching machine (from the programmed instruction methodology period of research). The student can receive 19 lines of text per frame and then must input a response on line 20. Sequencing of curriculum is performed by moving from page to page of material. In this mode of operation, the computer merely acts as a page turner. No really new innovations are apparent in this effort. No data as to the effectiveness or use of the FORTRAN course was available, however, many flowcharts and examples of student execution were included in the system documentation where the information reported here was obtained.

Another effort to teach FORTRAN programming is being conducted at the U. S. Naval Weapons Laboratory in Dahlgren, Virginia. A CAI system has been developed to drive the instructional program that teaches FORTRAN [11]. This system provides student record keeping facilities, response analysis procedures and automatic restart facilities so that a student can signoff of the course and be restarted at the position where he quit, the next time he signed onto the system. Also included are optional and mandatory review of curriculum materials that gave the student trouble when he took the course.

Output from the system appears on IBM 2250 cathode-ray tube graphic display terminals. Only two units are used at the present time. An innovation that sets this effort apart from the others discussed above is that the CAI system is interfaced with a FORTRAN interpreter called DISPLAYTRAN. The interpreter is used in conjuction with the CAI system to analyze student responses. In other words, when the student inputs a response in FORTRAN, DISPLAYTRAN processes the response and gives any necessary diagnostics and then the CAI system will present an output line such as "TRY AGAIN" or "VERY GOOD." The student is given four chances to answer correctly to a question. After that, the CAI system records the fact that a correct response was never given so that the particular curriculum being taught will be reviewed at a later time. Although no data relative to course effectiveness was available at this time, results from testing of this course are expected by September, 1970.

The methods used to teach the BASIC programming language at Dartmouth College constitute yet another use of the computer to assist instruction in programming. The learning situation begins with the student sitting in front of a teletype and reading a manual entitled, <u>Beginning BASIC, Everyman's Introduction To The Teletype And The BASIC Language</u> [5]. The manual instructs the student in the use of the teletype and then presents programming exercises of graduated difficulty. These program exercises are a student-controlled sequence of programs called BASIC [22]. The student must successfully complete one program before he can proceed to the next one in sequence.

BASIC is a "conversational" language for use on a teletype. Both arithmetic and file creation and manipulation routines are provided within BASIC. By progressing through the BASICT sequence the student learns the BASIC language. A student may terminate a session at the teletype anytime that he has received the name of his next program assignment. When he returns to the teletype, the student may type "NEW" and begin writing the new program assignment. Although no actual data was presented in the documentation of the efforts at Dartmouth to teach BASIC, it was stated that BASICT has been successfully used by adults and school children in grades 7 to 12 in learning BASIC.

A facility whose philosophy is similar to the JUDGE mode discussed in the CAI effort at Brighton College is available at Dartmouth in the form of a TEACH program [19]. The instructor writes a "skeleton" BASIC program that can be used to determine the correctness of a student's program under a variety of conditions. When the student is ready to check his program, he types TEST. The TEACH compiler builds a new program which consists of the student's program in a modified form and the skeleton program written by the instructor. The TEACH skeleton program sets up the data and then the student's program is executed and messages furnished by the TEACH program are output.

SCOOP (Student - Controlled On-line Programming) is a system used by International Computers and Tabulators Limited to give the student extensive real-time facilities for testing small programs written in the

PLAN language [29]. Many of the facilities reported in this project are similar to those found in using BASIC. The student may type in a small program, request execution of the code beginning at a particular point, request results from execution and edit the code on the basis of the results. This effort may be considered a use of CAI in the same sense that the project to teach BASIC was considered.

A number of papers concerning programmed instruction as well as computer-assisted instruction have been published by a research team at the Human Resources Research Organization in Alexandria, Virginia. Their description of a self-instructional course in basic computer programming provides some interesting ideas even though the course is presented through a programmed instruction manual. The student progresses linearly through the course and learns a language called FIELDATA which is used on the U. S. Army's MOBIDIC computer [4]. Such concepts as command, address, accumulator, memory location, input and output are presented in conjunction with sample programming problems.

Current work in CAI at the Human Resources Research Organization (HumRRO) is embodied in Project IMPACT (Instructional Model/Prototypes Attainable in Computerized Training) [33]. Project IMPACT is described as a "total systems development effort designed to provide the Army with an effective, efficient and economical CAI system." [32] The project is divided into four phases or development cycles which are as follows: Hardware, software, instructional content and instructional decision model.

Work is currently being performed in all four phases of the project. A significant innovation in this effort is the use of a theoretically-based instructional decision model unlike any used previously. An additional innovation is the use of Valid Confidence Testing (VCT) to allow the student a chance to indicate the confidence he has in his answer. The decision-making strategy to be used in course presentation entails the avoidance of prompting or confirmation when the student is working well. Just enough encouragement is given to keep the student progressing satisfactorily through the curriculum. Also of interest in the HumRRO effort is the use of a new and powerful data structure which can be used to develop more effective strategies as study progresses. It is difficult to estimate the impact that Project IMPACT will have on CAI at this time; however, it may well provide some profound findings. COBOL is the language which is to be taught by the CAI system once it has been developed. Project IMPACT is still in a development stage and therefore nothing can be said concerning the effectiveness of the CAI course.

A CAI system called TEACH[1] is currently being used to teach elementary programming at the Massachusetts Institute of Technology [14]. The chief objective of this effort was to teach programming and not just a particular programming language. A programming language called UNCL is used, however, to convey many of the ideas that cannot be presented without a language. The philosophy underlying the development of UNCL was much the same as that for the ELASTIC language, that is, pedagogical considerations influenced the design and structure of the UNCL language. UNCL is an interactive language that resembles the JOSS language in many respects.

The TEACH system was written in MAD and SLIP. It executes under the Compatible Time Sharing System (CTSS) on an IBM 7094 computer. The output terminal used by the student is a teletype or IBM selectric. Features

[1]This TEACH system is to be distinguished from the TEACH programs explained in the CAI effort to teach BASIC at Dartmouth.

included in TEACH are self-paced student sequencing through curriculum, student record facilities, a restart procedure, and on-line interaction with the language being learned. Course material to teach programming is written in a language called AGES. The AGES interpreter surrounds the UNCL interpreter and is used to present text to the student and analyze responses with the help of the UNCL interpreter. Programs which present the curriculum are called "scripts." These scripts control the information typed to the student, which in reality constitute a text in programming. A script can turn control of the UNCL interpreter over to the student so that he can write assigned exercises. Within the TEACH system, however, there is no facility to determine exactly how the student uses the UNCL interpreter. That is, the student can write any program or exercise he wishes regardless of whether it is the current assigned problem or not.

Although TEACH has been used to give instruction to a number of students, no tests have been made regarding the validity or effectiveness of the teaching system. According to the report of this system in the March, 1970 issue of Communications of the Association for Computing Machinery, interviews with students who successfully completed the course indicated that they had absorbed a significant amount of material.

There are two CAI efforts to teach programming being conducted at Stanford University. The first of these to be discussed has as its major goal, the development of a CAI system to teach programming [9]. The language used to teach programming is called AID. AID (Algebraic Interpretive Dialogue) is an interactive language which is based on JOSS. It was originally implemented by Digital Equipment Corporation for execution on PDP computers. While still in a development and revision stage, the instructional system consists of a lesson preprocessor and lesson driver. The preprocessor translates a lesson into suitable form for input to the driver. The driver, using the lesson directory prepared by the preprocessor, presents a lesson to the student. The student response analysis routines include exact match, keyword match and numerical equivalence.

The curriculum to teach AID is linearly presented; however, the student can skip sections and jump ahead as he wishes. There are a total of 30 lessons in the course and an average lesson in the AID curriculum has 21 problems. In addition to taking the CAI course the student uses a specially-prepared student manual that contains homework assignments, extra-credit problems and outside readings as well as an AID reference manual. Reimplementation of the CAI system and the curriculum material for a new computer has significantly slowed the progress of this effort. No data relative to course effectiveness has been published to date.

The second endeavor at Stanford concerns the teaching of programming using the languages SIMPER and LOGO [10]. SIMPER is a pseudo-assembly language designed by Paul Lorton at Stanford. The language is very simple (20 operations) and is taught during the first half (37-1/2 hours) of a high school semester. LOGO, which is a list processing language designed by Bolt, Beranek and Newman, is taught during the second half of the course. As in the AID project, the student is furnished with a student manual and language reference manuals. The curriculum material is presented using the instructional system developed in the AID project. There are 38 SIMPER lessons and 19 LOGO lessons at present; however, the final course in LOGO will require 49 lessons.

Preliminary testing of the CAI course took place at Woodrow Wilson High School in San Francisco. A PDP8i computer was used to drive twelve teletypes. It was stated that several students finished the 38 SIMPER lessons and 19 LOGO lessons and that most of the students finished SIMPER and 5 to 10 lessons on LOGO. Based on this testing, revisions in the curriculum are now being made. The primary purpose of most of the revisions is to attempt to distinguish between the teaching program which presents lessons about the language being learned and the language which the student learns and in which he writes programs. Other revisions include renaming of SIMPER instructions to avoid confusion and addition of more sophisticated instructions to the instruction set. This course is also being reimplemented for a new computer as is the AID course.

The only results reported from this preliminary testing of the SIMPER and the LOGO course was that a significant degree of attitudinal and motivational gains had been noted in the students who participated in the CAI experience.

In summarizing the projects presented above, it can be noted that the computer can assume varying roles in the instructional process to teach programming. The simplest role finds the computer simulating the actions of a teaching machine. In another instance the computer can be used to execute programs or code segments in the language being taught, that is, the computer is simulating itself. The role of program checker, grader, and record keeper is sometimes assumed by the computer, and finally the computer can be used to present the curriculum material directly to the student in branching, tutorial mode. All of the projects mentioned previously used the computer in one or more of these roles.

As stated previously one of the most significant findings of the literature search was the fact that very few of the projects reported any extensive testing or evaluation of the instructional program. The conducting of just such tests and evaluations of an instructional program to teach programming is one of the expressed purposes of the effort reported here.

CHAPTER II

Instructional Program

The purpose of this chapter is to explain the design, development and pertinent elements of the instructional program to teach the ELASTIC programming language. The version of the program used in the formative evaluation will be discussed here while the current version of the program used in the summative evaluation will be discussed more fully in Chapter IV.

The structure of ELASTIC had some bearing upon the design of the instructional program. It is for this reason that a brief description of the ELASTIC computers and language is presented below. It should be remembered throughout this discussion that the ELASTIC system is a FORTRAN program which simulates a computer with desired characteristics.

Description of ELASTIC

At present there are four "ELASTIC Computers" [17]. The ELASTIC1 computer is a binary, sequential machine which has 4096 words of memory. Each word is 36 bits long. Fixed-point arithmetic performed modulo $2^{36}-1$ is available in ELASTIC1. The ELASTIC1 computer has A and Q registers, both 36 bits long. The instruction word format for an ELASTIC1 instruction is illustrated in Figure 1.

The ELASTIC1 computer processes the basic instructions of a typical assembler language, that is, basic information transfer instructions (LOAD and STORE), basic arithmetic instructions (ADD, SUBTRACT, MULTIPLY and

DIVIDE), basic input/output instructions (READ and PRINT) and basic transfer of control instructions (UNCONDITIONAL BRANCH and CONDITIONAL BRANCH).

operation code 6 bits	18 zero bits	address 12 bits
35 — 30	29 — 12	11 — 0

Figure 1

ELASTIC1 Instruction Word Format

The ELASTIC symbolic instruction consists of four fields: LOCATION, OPERATION CODE, ADDRESS, and REMARKS. An ELASTIC program is input to the ELASTIC system on cards, one instruction per card. The ELASTIC symbolic instruction format is illustrated in Figure 2.

column 1 — 6	7	8 — 10	11	12 — 28	29	30 — 80
Location field		Operation field		Address field		Remarks field

Figure 2

ELASTIC Symbolic Instruction Card Format

The ELASTIC2 computer processes an expanded subset of the ELASTIC language. ELASTIC2 retains all of the capabilities of ELASTIC1 as well as the instructions for manipulating index registers. The instructions which manipulate the index registers are typical: LOAD INDEX and STORE INDEX, ENTER INDEX and INCREASE INDEX and finally CONDITIONAL BRANCH ON INDEX. BO is always zero in ELASTIC. In ELASTIC2, the contents of designated index registers may be used to modify the address field of any instruction. This facility in effect produces new instructions which are composed of an

ELASTIC1 instruction coupled with an index register designation. Also added in ELASTIC2 is the facility for reading more than a single value from a card and printing more than one value on a line. This is accomplished by a two-address logic scheme which causes the instruction word format in ELASTIC2 to appear as illustrated in Figure 3.

operation code 6 bits	secondary tag 3 bits	primary tag 3 bits	secondary address 12 bits	primary address 12 bits
35 30	29 27	26 24	23 12	11 0

Figure 3

ELASTIC2 Instruction Word Format

Unused portions of particular instruction words are filled with zero bits.

In ELASTIC2, expressions including +, -, *, /, symbols and constants may appear in the address field of a symbolic instruction. Expressions are scanned strictly left to right by the assembler.

The ELASTIC3 computer retains all capabilities of ELASTIC1 and ELASTIC2. ELASTIC3 also processes SHIFT instructions, LOGICAL arithmetic instructions and input/output instructions which process alphanumeric information. In ELASTIC3 the capacity for creating octal constants and Hollerith constants is added. Also added in ELASTIC3 is the ability to designate the current instruction address, denoted by $, within the address field of an instruction. The instruction word format for ELASTIC3 is the same as that for ELASTIC2. (See Figure 3.)

What might be considered ELASTIC4 is the capability to define and call macros in an ELASTIC program [16]. The word format in ELASTIC4 is identical to that for ELASTIC2 and ELASTIC3. The instructions which allow definition and reference to macros as well as conditional assembly instructions are added to the ELASTIC language in the ELASTIC4 computer.

The instruction set constituting the ELASTIC programming language appears in Appendix D. A review of the instruction set should bring the realization that ELASTIC is "typical" and that a large number of basic computer concepts may be presented utilizing these instructions.

Objectives

The terminal objective of the instructional program as stated in behavioral terms is presented below.

1. Given a problem he has never seen before which can be solved by means of an ELASTIC program, the student will be able to:

a. organize and plan the programming solution

b. code and debug the corresponding ELASTIC program.

This objective will be met if the program written by the student will execute properly on the computer, providing a correct solution to the problem posed.

It should be noted that the ELASTIC language is merely an instructional tool and is therefore relatively restricted and simplified in nature. A review of the language and its capabilities will substantiate this fact. This delimits the objective to a class of relatively simple but representative assembly language problems.

Prerequisites and Constraints

The prerequisite for taking this course was satisfactory completion of Computer Sciences 404G or another equivalent introductory computer programming course. Although the instructional program required little prior programming background, the prerequisite for enrolling in Computer Sc.ences 310, of which this program is a part, requires that the student have had previous programming experience.

The primary constraint involved with learning ELASTIC by computer-assisted instruction would seem to be the time factor both with respect to duration or length of the course and with respect to scheduling of students to take the course. A constraint vital to the design of the program is the typical display devices available. A teletypewriter was used but it would have been desirable to have had a terminal capable of providing a much faster output rate. (An IBM 1050 typewriter was used as the output device with the original version of the program but this device also had serious drawbacks.) Another constraint which may or may not be present is the availability of an adequate number of output terminals to service the student population.

Task Analysis

A flowchart of the task analysis for the instructional program to teach ELASTIC appears in Figure 4 below. This task analysis was performed with the primary objective in mind: the student will obtain the ability to organize and plan the solution to problems using the ELASTIC programming language.

The flowchart is to be interpreted beginning at the bottom of the page and progressing upward. The development of the subject matter is arranged in hierarchical form in such a way that the interrelationship between particular concepts and skills is graphically illustrated by the position of each enclosure and by the interconnecting lines and directional arrows.

Development

Two versions of the instructional program to teach the ELASTIC programming language have been developed. The original version or what might be considered the prototype version of the CAI course was used in the

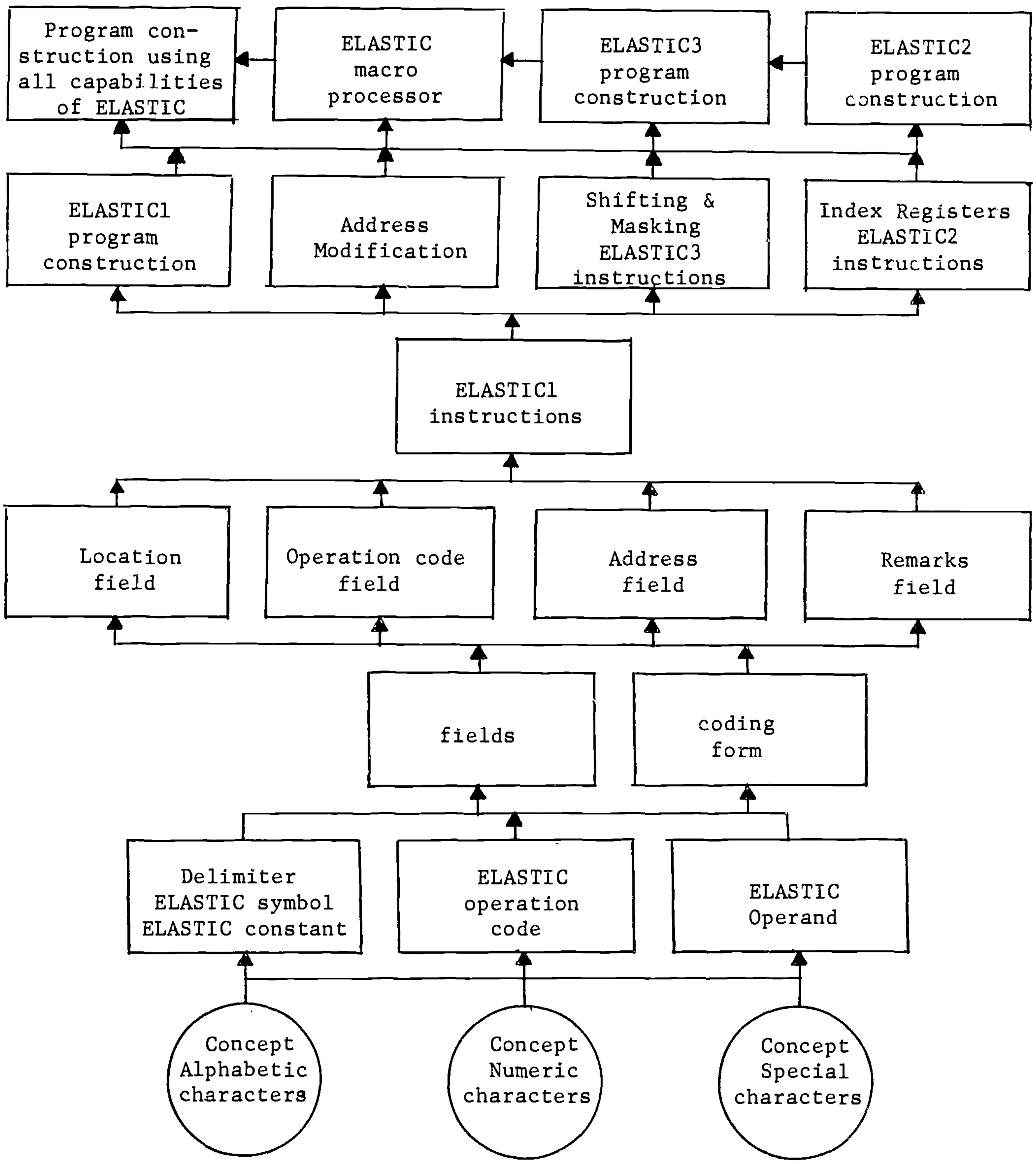


FIGURE 4

Task Analysis

formative evaluation reported in Chapter III. Results from this evaluation prompted a revision and reimplementation of the program. It is this revised or final version that was used in the summative evaluation reported in Chapter V. Both implementations will be discussed below.

The prototype version of the instructional program was written in IBM's COURSEWRITER I language. The course was executed on an IBM 1440 computer that used IBM 1050 typewriter terminals as the system/student interface devices. The 1440 system was made available through the University of Texas Computer-Assisted Instruction Laboratory. The CAI course was structured as one large program. Various sections of the course could be referenced by using labels which were assigned to numerous instructions throughout the program. The student could use the labels to branch around in the course and to review previous material taken. The list of labels given to the student appears in Appendix A. Sequencing through the curriculum was principally linear, however, as mentioned above the student could vary the linear sequence of presentation. A linear sequencing scheme was chosen primarily because the student normally has no prior knowledge of ELASTIC and thus each student should receive basically the same material.

A flowchart for the instructional program may also be found in Appendix A of this paper. It is from this flowchart that the prototype version was designed. As illustrated in the flowchart, in a normal situation the student was given two or three chances, depending on his responses, to give the correct answer to each question asked, after which he was given the correct answer and requested to type it into the terminal keyboard. The development of the course was logically divided in the same manner as the ELASTIC language, that is, the curriculum for ELASTIC1 was coded and debugged before the development of the segment for ELASTIC2 was begun and so on. Using this approach facilitated preliminary testing of the program. When finally completed the instructional program required approximately 9.7 hours at the terminal for successful completion by the average student.

Two quizzes were included in the course, one after ELASTIC1 had been presented and the other after the material on ELASTIC3 was given to the student. After taking each quiz, the student was given the choice of seeing his grade for the quiz. Whether or not he chose to see the grade, he was given the opportunity to see the correct answers so that he could determine the questions that he missed and why. The student was strongly encouraged to go back and review those sections in the course that he did not understand after discovering the correct answers to questions he had missed.

The COURSEWRITER system used on the IBM 1440 computer provided both a restart procedure for continuing in a course from session to session and facilities for maintaining a student record file. Each student that took the CAI course on ELASTIC was assigned a unique password or student number. Everytime the student signed onto the course he used his individual password. Using this number, the COURSEWRITER system kept track of where the student was in the course at all times. When the student terminated a session the system would record the position in the course so that when the student signed onto the course again, he was started at the same position in the course where he stopped in the previous session. At the discretion of the course author the COURSEWRITER system would record a number of data items about each student as he took the course. This information included a permanent record of each response that each student made to each question in the course, a record of the position of every student in the course at any given time and a listing of the dates and total time that each student had used the instructional course. All of the data items enumerated above were collected during the formative evaluation.

The data and results recorded by the COURSEWRITER I system during the formative evaluation were used to develop an improved version of the instructional program. This final version was coded in the PICLS language (developed at Purdue University) [25] and was executed on the Control Data Corporation (CDC) 6600 computer system at the University of Texas at Austin. The CAI course was executed by the PICLS system which in turn was executed in the conversational mode of the RESPOND time-sharing system which is available on the 6600 computer [24]. The output terminal used to interface the student and the CAI program was a teletypewriter in this case. It should be mentioned that a Data Point 3300 cathode-ray tube terminal could be and was used periodically in the testing phases of this version of the program. An overview of the PICLS language is presented in Appendix E of this paper while a description of the RESPOND time-sharing system appears in Appendix F. A more complete description of both the hardware and software environment in which the final version of the instructional course resided may be found in Chapter IV. The remainder of this section is concerned with discussing the final version of the course and how it differed from the prototype version.

The final version of the instructional program still reflected the basic structure and design of the prototype; however, a number of improvements, refinements and more powerful facilities were incorporated into the instructional situation. Improvements included the removal of ambiguities from statements and questions, and clarification of vague statements throughout the course. Incorporation into the course of more elaborate answer processing functions and facilities that were available in the PICLS author language made possible the processing of a much wider range of student responses than had been used in the COURSEWRITER version.

Due to system storage and time considerations the final version of the CAI course was organized as eighteen semi-independent subcourses. Each subcourse was further divided into two or three sections and each section consisted of from 50 to 350 coded PICLS author language instructions. An outline of the course structure as it appeared in the final version may be referenced in Appendix A. As a result of this restructuring, additional instructions were included throughout the course to enable the student to sequence himself through the curriculum. As in the prototype the student proceeded through the course material in a sequential, linear fashion. To review or move ahead, the student had facilities for branching from section to section and from subcourse to subcourse throughout the CAI course. Unlike the prototype version, however, branching to another frame further down in a section was possible but not easily accomplished.

Additional curriculum was written for almost all areas of the course, especially in the areas of the binary number review, programming techniques and program formation. Also appended were instructions for using the teletypewriter and procedures for entering responses.

Facilities for executing ELASTIC programs during a session at the teletype were added to the overall instructional situation in which the final instructional program was used. As anticipated the ability to interact with the ELASTIC language during the instructional session proved to be a boon to the student and a significant improvement to the instruction process. The student could exit from the PICLS system and immediately enter and execute an ELASTIC program. In fact he was prompted to do just this at various points throughout the course. The incorporation of on-line execution of ELASTIC programs contributed greatly to more complete and efficient use of the computer in the instructional process. A description of some of the problems involved in building and implementing the final form of the CAI course on the PICLS system is presented in the final section of Chapter III.

Computer-assisted instructional facilities have been available on the CDC 6600 computer since February 1, 1970. Due to this fact an "interim-experimental" version of the PICLS system was used to execute the final version of the instructional program. This experimental PICLS system had no restart or student record facilities implemented. Neither of these facilities proved to be critical (only inconvenient), however, in conducting the summative evaluation. The absence of a restart capability was compensated for by the student being made more fully aware of the structure of the instructional program than he would have otherwise been. By only terminating a session at the end of a section and through the use of a few PICLS commands, the student could "simulate" the restart procedure quite adequately. The student record facility was more difficult to "simulate" than the restart procedure explained above. The student was made responsible for keeping a log of all time spent and work done at the teletype. At the conclusion of the course the student was requested to turn in all teletype output that he had received throughout the CAI course in order that the various responses that various students made to each question could be reviewed. Statistics regarding the total computer time and RESPOND time used by each student were obtained through regular facilities provided to all users of the CDC 6600.

It is noteworthy that an improved PICLS system is currently being developed that includes both restart and student record capabilities as well as an improved author language. The problems encountered in reimplementation of the instructional program for execution under the PICLS system contributed to a realization of the improvements that were needed for an efficiently running CAI facility on the 6600.

In summary, an attempt to clarify the relationship between the two versions of the instructional program seems appropriate. The prototype version was a more-or-less skeleton program in that the basic core curriculum was present in a form that could later be improved and refined if the techniques of CAI to teach ELASTIC proved to be effective. The final version filled out the course by adding to the curriculum and improving the processing of student responses as well as incorporating an on-line ELASTIC program execution mode.

CHAPTER III

Formative Evaluation

Introduction

The use of computer-assisted instruction to teach computer programming is one of the most obvious and as yet unvalidated applications of this relative new pedagogy. This chapter describes an experiment that was conducted at the University of Texas at Austin in the Fall of 1968. The purpose of the experiment was to illustrate the practicallity of employing CAI techniques to teach assembly language programming. The evaluation described here might be classified as a summative evaluation or test of the final CAI product; however, the vast amount of data and feedback obtained from this study dictated extensive revision and modification to the instructional program. It is for this reason that the evaluation reported here is considered to be formative or developmental rather than summative. Description of the summative evaluation may be found in Chapter V.

While the evaluation study reported in this chapter uses some of the terminology and methodology of behavioral science research, the author recognizes the limited generality of the results. A "horse race" between a lecture version of a course and a CAI version based on the primitive author language, terminal device, and instructional strategies reported here makes the same amount of sense as a race between a horse and a Model T Ford, at least insofar as drawing generalizations about the intrinsic merits of horses and automobiles for rapid transportation. The concept of formative evaluation puts this study in context. If a CAI course can be shown to be as effective or almost as effective as a lecture course, and extensive data can be collected to focus on the limitations of the CAI method, then this data can be used to re-engineer the CAI system for greater effectiveness and lower costs. The forms and terms of behavioral research provide a powerful heuristic framework for hypothesizing empirical outcomes, describing methodology, and structuring data for useful analyses. It is for this reason that such forms and terms were employed in this report.

Hypotheses

The hypotheses to be tested in the experiment were as follows:

1. The CAI group will complete course instruction significantly faster than the lecture group;

2. The CAI group will make significantly more personal visits to the instructor;

3. The CAI group will ask significantly more advanced questions of the instructor in visits;

4. The CAI group will score significantly higher with respect to mean score on examinations;

5. The CAI group will score significantly higher on mean grade of computer programs written;

6. The CAI group will score significantly higher with respect to attitude about the course at its conclusion according to an attitude questionnaire to be administered.

Method

Two classes of students registered for Computer Science 310 were used in the experiment. One class served as the control group while the other was the test (CAI) group. One of the sections (classes) of students was scheduled to meet on Mondays, Wednesdays, and Fridays at 12 Noon (section 3) and the other section was scheduled to meet on Mondays, Wednesdays and Fridays at 3 P.M. (section 5). Section 3 was arbitrarily chosen to be the CAI group while section 5 served as the lecture group. Students got into a particular section by registering for one of the eight available sections of Computer Sciences 310 that they could attend. The control group (lecture group) received instruction through formal classroom lectures, while the test group was self-paced and utilized a CAI course for their formal instruction. The CAI course was administered through use of an IBM 1440 computer system in the CAI Laboratory at the University of Texas at Austin. The assembly language taught to both classes is named ELASTIC. ELASTIC is a language that was developed at the University of Texas at Austin to be used exclusively as a teaching tool for instruction in assembly language programming.

At the conclusion of the experiment, a comparison was made between the groups according to the following criteria: examination grades, program grades, term project grades, responses to an attitude questionnaire, number of personal visits to the instructor's office, and ratings of conversations between the students and the classroom instructor.

The classes that participated in the experiment were enrolled in Computer Sciences 310 at the University of Texas at Austin. The instructor for both classes was Fred Homeyer. For the first three weeks of the semester, both classes were given formal lectures on general computer concepts. At the conclusion of this period a test was given to both classes. It was hoped that there would be no significant difference in the mean test score of both classes so that t-tests could be used to test for significance between means on subsequent data to be collected during the semester.

For the remainder of the semester, the lecture group received formal lectures and the test group received instruction through a specially designed computer-assisted course described in Chapter II. Both classes were given two hour examinations and a final examination over the fundamentals and use of the assembly language they had learned. During the semester, each student developed and tested a total of eight computer programs. In addition to these programs, every student completed a term project assignment. The term project was concerned with designing and implementing an assembler program which assembled symbolic code for a hypothetical computer. The program assignments and examinations which each student was assigned may be referenced in Appendix C. Figure 5 illustrates the experimental design used in the formative evaluation. The lecture group consisted of 16 students while the test group numbered 24.

To test the first hypothesis, the average time spent by a student taking the CAI course was compared against the average time spent per student in formal classroom lectures for the control group. A log of instruction time for both classes was kept throughout the semester.

To test the second hypothesis, the average number of visits per student in both groups was computed and these figures were compared and tested for significant difference. A diary of visits for each student was kept throughout the semester.

To test the third hypothesis, the questions asked by the students in personal visits with the instructor were rated according to the scale: elementary, intermediate and advanced. The average rating of questions for each class was computed. The instructor should have been as objective in the question rating as possible, however, since there was no assurance of this the results of this test should be considered in this light.

Variable	Test Group(CAI)	Lecture Group
Instructor	Fred Homeyer	Fred Homeyer
Tests	Concepts test ELASTIC1 test ELASTIC3 test Final exam	Concepts test ELASTIC1 test ELASTIC3 test Final exam
Time spent on programs	Date started and date completed to be included in each program documentation turned in for each student	
Programs	Eight programs and a term project	
Questions	Questions in visits with instructor rated according to the scale: advanced, intermediate, elementary	
Number of office visits with instructor	Log kept by instructor throughout the semester	
Text	"The ELASTIC Programming Language Reference Manual"	

Figure 5

Formative Evaluation Design

In order to test hypothesis 4, the mean examination scores for each class was computed and compared. The standard deviation for each test was also computed for each group. From the means and standard deviations, t-tests for testing of significant difference between means were computed.

To test hypothesis 5, the mean grade and standard deviation for each program for both classes was computed. The program assignments were documented by the students according to an outline entitled, "Procedure for Program Documentation." A copy of this outline appears as the last pages of Appendix C. A student assistant graded the program assignments using the documentation form with particular sections of the form weighted as follows: purpose of the program - 10%; use of the program - 10%; information on subroutines - 20%; description of processing procedure - 20%; achievement of purpose and efficiency - 20%; flowchart and listing of the program - 20%. The student assistant who graded the programs was aware that a study was being performed and he also was aware of which students were in which group. For these reasons, possible bias on the part of the grader should be considered when examining the results of testing this hypothesis. The values were then tested for significance using t-tests and F-tests were used to test for a difference in variances.

To test hypothesis 6, an attitude and course evaluation questionnaire was completed by each student at the end of the semester. A copy of the questionnaire appears in Appendix H. The CAI group answered some additional questions concerning the quality and applicability of specific portions of the CAI course which they took. The responses for most items on the questionnaire could range in value from 1 to 9. The average response and standard deviation for each item was computed and tested for a significant difference between groups.

Results

The results of the experiment will be presented in order by hypothesis as listed in the second section of this chapter. Prior to presenting the results, however, a "profile" of each student is presented in Tables 1 and 2. Table 1 contains the profiles for students in the lecture group while Table 2 lists the profiles of students in the CAI group. The profile gives the following information for each student: student number, sex, classification, major, number of previous computer courses taken, grades in previous courses, previous experience with CAI, languages in which the student can program, number of office visits made during the semester, average rating of questions asked by the student

TABLE 1

STUDENT PROFILE (LECTURE)

Formative Evaluation

Number	Sex	Class.	Major	Previous Computer Course	Grade	Previous exp. with CAI	Language Used
1	F	soph	Psy.	1	C	no	ALGOL
2	M	sr	Math	1	D	no	ALGOL
3	F	jr	Math	1	B	no	ALGOL
4	F	sr	Math	1	B	no	ALGOL
5	M	sr	Math	1	A	no	FTN
6	M	jr	Math	1	B	no	FTN
7	M	soph	Math	1	B	no	ALGOL
8	M	jr	Math	1	B	no	ALGOL
9	M	sr	Math	1	B	no	ALGOL
10	M	soph	Math	1	A	no	FTN
11	M	jr	EE	1	C	no	FTN
12	F	jr	Math	1	C	no	FTN
13	M	sr	Math	1	B	no	FTN
14	M	sr	EE	1	A	no	FTN
15	M	jr	Math	2	A & B	yes	FTN
16	M	grad	Psy.	1	A	no	ALGOL

TABLE 1--Continued

Number	No. of office visits	Question Rating	Program avg. grade	Quiz avg. grade	Term project grade	Course grade
1	1	2	88.14	87.33	80	86.64
2	0	0	69.07	70.00	65	71.92
3	8	1.6	91.43	81.67	88	87.53
4	7	1.4	97.57	83.33	88	88.67
5	1	3	93.79	91.67	90	92.84
6	1	2	93.93	81.67	80	85.88
7	4	1.5	41.29	72.33	80	68.09
8	0	0	89.71	84.67	80	85.91
9	0	0	87.71	78.00	78	82.91
10	0	0	90.57	89.33	88	89.57
11	0	0	89.93	75.00	80	81.48
12	7	1.7	85.79	67.33	85	80.94
13	2	2.5	94.00	81.33	85	88.40
14	0	0	86.07	81.33	85	85.82
15	2	2	88.36	75.67	88	84.01
16	2	3	94.29	83.33	95	90.89

TABLE 2

STUDENT PROFILE (CAI)

Formative Evaluation

Number	Sex	Class.	Major	Previous Computer Course	Grade	Previous exp. with CAI	Language Used
1	F	sr	Math	1	A	no	FTN
2	M	grad	Psy.	1	A	no	FTN
3	M	sr	EE	1	A	no	FTN
4	M	sr	Bus.	2	A & B	no	FTN
5	M	sr	Acct.	2	A & B	no	FTN
6	M	jr	Stat.	4	C	no	FTN
7	M	grad	Fin.	1	A	no	FTN
8	F	jr	Math	1	A	no	ALGOL
9	F	soph	Math	1	C	no	FTN
10	F	sr	Math	1	B	no	FTN
11	F	jr	Math	1	C	no	ALGOL
12	F	grad	C.S.	–	–	no	ALGOL
13	F	soph	Math	2	A & A	no	FTN
14	M	sr	Pet.E.	1	A	no	ALGOL
15	M	jr	Physics	1	A	no	FTN
16	M	sr	EE	1	B	no	FTN
17	F	jr	Math	1	B	no	FTN
18	F	sr	Chem.	1	B	no	ALGOL
19	M	sr	Math	1	A	no	FTN
20	F	jr	Math	2	A & A	no	FTN
21	F	jr	Math	1	A	no	FTN
22	M	sr	Bus.	1	B	yes	FTN
23	F	grad	Stat.	1	A	yes	FTN
24	M	sr	Eco.	1	A	no	ALGOL

TABLE 2--Continued

Number	No. of office visits	Question Rating	Program avg. grade	Quiz avg. grade	Term project grade	Course grade
1	5	2	98.57	92.33	90	94.67
2	2	3	93.43	83.67	88	86.53
3	0	0	97.57	88.33	85	92.17
4	2	2	84.36	76.67	88	82.11
5	1	2	96.43	84.00	90	89.33
6	0	0	86.36	69.00	85	77.01
7	0	0	92.93	77.00	80	82.98
8	3	2.2	93.57	74.67	85	83.87
9	2	1.5	85.79	76.00	70	75.94
10	4	2	90.21	83.33	85	87.26
11	2	2.5	94.43	81.67	85	88.03
12	2	1.5	93.14	73.00	75	79.84
13	1	3	87.50	78.67	80	82.65
14	1	3	92.71	81.33	100	92.01
15	0	0	84.50	94.67	85	89.55
16	0	0	80.43	83.00	85	82.23
17	1	3	96.93	90.00	90	91.08
18	0	0	53.79	81.67	70	67.04
19	4	3	99.07	98.33	100	98.62
20	2	2.5	98.00	82.33	88	90.10
21	4	1.8	98.57	82.33	88	89.67
22	1	2	68.71	72.33	80	71.51
23	3	2.2	94.14	83.67	88	89.74
24	0	0	97.86	87.33	88	91.16

(1=elementary, 2=intermediate, 3=advanced), the average of the grades made on programs, the average grade made on examinations, term project grade and final course grade.

It should be noted that all tests of the hypotheses listed previously used a 0.05 level of significance. Student's t-tests were used for testing to determine if there was a significant difference between the arithmetic means of the two groups and variance ratio F-tests were used to test for significant difference in variances when the separate program assignments were considered.

Hypothesis (1): The CAI group will complete course instruction significantly faster than the lecture group.

Results:

	Lecture Group	CAI Group
Number of students	16	24
Total time spent	24 hours of formal classroom lecture	234.91 hours of computer time
Average	24 hours/student	9.7 hours/student

Conclusion: Hypothesis accepted; the CAI group completed course instruction more than twice as fast.

Hypothesis (2): The CAI group will make significantly more personal visits to the instructor.

Results:

	Lecture Group	CAI Group
Number of students	16	24
Total number of visits	35	40
Average number of visits per student	2.18	1.66
t-test	0.86	

Conclusion: Hypothesis rejected; the CAI group had fewer visits per student.

Hypothesis (3): The CAI group will ask significantly more advanced questions of the instructor in office visits.

Results: (The questions were rated as follows: 1 = elementary, 2 = intermediate, 3 = advanced)

	Lecture Group	CAI Group
Number of students	16	24
Total number of visits	35	40
Total weighted value of all questions asked	64	91
Average quality rating per question	1.82	2.225
Standard deviation	0.61	0.64

Conclusion: Because the ratings of question quality were obtained by the experimenter using a subjective rating scale, no statistical test was performed. Note, however, that according to the data presented, there is a substantial difference in the quality of questions asked by students in the CAI group when compared to students in the lecture group. The reader may judge for himself concerning the hypothesis by examining the raw data presented in Tables 1 and 2.

Hypothesis (4): The CAI group will score significantly higher with respect to mean score on examinations.

Results:

		Lecture Group		CAI Group
Quiz #1:	Average grade	87.06		89.67
	S. D.	4.46		5.70
	t-test		-1.50	

Conclusion on Quiz #1: Using a 0.05 level of significance, there is no significant difference between the means. (Both groups had formal lectures up to this point in the semester, therefore, the assumption can be made that the groups are equal at the beginning of the experiment based on the data above.)

Quiz #2:	Average grade	84.00		82.92
	S. D.	8.68		7.60
	t-test		0.41	

Conclusion on Quiz #2: Using a 0.05 level of significance, there is no significant difference between the means of the two groups.

Quiz #3:	Average grade	69.69		74.33
	S. D.	13.19		12.73
	t-test		-1.09	

Conclusion on Quiz #3: Using a 0.05 level of significance, there is no significant difference between the means of the two groups.

Final Exam:

Average grade	89.00		84.42
S.D.	3.92		10.58
t-test		1.62	

Conclusion on Final Exam: Using a 0.05 level of significance, there is no significant difference between the means of the two groups.

Conclusion of hypothesis: Reject the hypothesis; there is no significant difference in the performance of the two groups with respect to examination grades.

Hypothesis (5): The CAI group will score significantly higher on mean grade of computer programs written.

Results:

	Lecture Group		CAI Group
Program #1			
Average grade	90.19		94.00
S. D.	6.15		4.16
t-test		-2.28*	
F-test		2.18	

Conclusion on Program #1: There is a significant difference between the means at the 0.05 level. The CAI group scored higher on program #1. The variances could not be considered different at the 0.05 level of significance.

	Lecture Group		CAI Group
Program #2:			
Average grade	90.69		94.42
S. D.	8.76		3.83
t-test		-1.79	
F-test		5.23*	

Conclusion on Program #2: There is no significant difference between the means of the two groups at the 0.05 level of significance. The variances are unequal at the 0.05 level. The CAI group was less variable.

---

* significantly different at the 0.05 probability level

Program #3:

Average grade	89.12		92.92
S. D.	8.29		6.15
t-test		-1.62	
T-test		1.82	

Conclusion on Program #3: There is no significant difference between the mean or the variances using a 0.05 level of significance.

Program #4:

Average grade	82.31		87.92
S. D.	9.29		11.92
t-test		-1.55	
F-test		0.06	

Conclusion on Program #4: There is no significant difference between the means or the variances of the two groups. Both tests used a 0.05 level of significance.

Program #5:

Average grade	88.94		87.75
S. D.	10.51		19.41
t-test		0.22	
F-test		0.29	

Conclusion on Program #5: There is no significant difference between the means or the variances of the two groups. Both tests used a 0.05 level of significance.

Program #7:

Average grade	81.56		81.83
S. D.	22.81		26.62
t-test		-0.03	
F-test		0.73	

Conclusion on Program #7: There is no significant difference between the means or the variances at the .05 level of significance.

Program #10:

Average grade	84.44		92.79
S. D.	23.39		8.62
t-test		-1.55	
F-test		7.37*	

Conclusion on Program #10: There is no significance between the means but the variances are unequal at the 0.05 level of significance. The CAI group was less variable.

Program #11:

Average grade	87.50		94.79
S.D.	24.71		7.96
t-test		-1.31	
F-test		9.64*	

Conclusion on Program #11: There is no significant difference between the means of the two groups at a 0.05 level of significance; however, the variances are unequal at the same level. The CAI group was less variable.

Program Average

Average grade	86.35		89.95
S.D.	13.58		10.51
t-test		-.919	
F-test		1.66	

* significantly difference at the .05 probability level

Conclusion on Program Average: There is no significant difference between the means or the variance of the program average of the two groups at the 0.05 level of significance.

Term Project:

Average grade	83.44		85.33
S.D.	6.60		7.07
t-test		-0.83	
F-test		0.87	

Conclusion on Term Project: There is no significant difference between the means or variances of the two groups at the 0.05 level of significance.

Conclusion on hypothesis: Reject the hypothesis; the groups performed approximately the same on program grades. In the cases where there was a significant difference in means for a particular program, the CAI group scored higher than the lecture group. In the cases where there was a significant difference in variances in program grades, the CAI group was less variable than the lecture group.

Hypothesis (6): The CAI group will score higher with respect to attitude about the course at its conclusion according to an attitude questionnaire to be administered.

Results: The results of this portion of the experiment appear in Table 3. The questionnaire that was answered by each student appears in Appendix H. The majority of the questions could range in value from 1 to 9 with 9 being the more positive response. For the exceptions, see the sample questionnaire in Appendix H.

Conclusion: Reject the hypothesis; of the 39 questions analyzed (questions 1 through 39), there were only three questions that showed a significant difference between the mean answer of each group at the 0.05 level of significance. It can be concluded that the attitudes of the two groups, at least as measured by the questionnaire used here, were basically the same.

TABLE 3

DATA FROM ATTITUDE QUESTIONNAIRE - FORMATIVE EVALUATION

Question	Mean Ratings		*t*	Standard Deviations	
	Lecture	CAI		Lecture	CAI
1. General benefit of this course to most individuals.	6.00	6.43	-.58	2.13	1.81
2. General approach of the instructor to the course and subject matter.	7.55	7.48	.13	1.10	1.78
3. General helpfulness and value of the textbook used in the course.	7.00	6.57	.57	1.28	2.24
4. Overall impression of the instructor when compared with ideal instructor.	7.45	6.76	1.49	.99	1.31
5. Fairness of the contents of tests in this course.	6.45	6.24	.32	1.30	1.92
6. General benefit of homework in the course.	5.91	6.10	-.29	1.78	1.63
7. Fairness of grading in this course.	7.73	6.29	2.43*	1.14	1.72

* Significant at the .05 probability level

TABLE 3--Continued

8. Availability of equipment in labs for this course.					
	2.36	4.71	-2.36*	1.77	2.93
9. Realization of goals in the course when compared with goals you set.					
	6.45	6.33	.14	2.31	2.19
10. Instructor's personal interest in students.					
	7.27	6.38	1.41	1.35	1.79
11. Daily preparation of instructor.					
	8.18	8.29	-.31	.83	.88
12. General benefit of lab in learning the information in this course.					
	5.64	6.76	-1.36	2.53	1.92
13. Adequacy of the amount of reading required for the course.					
	6.09	6.00	.12	1.56	2.12
14. Clarity and understandability of tests in the course.					
	5.91	5.57	.42	2.23	2.04
15. Challenge of tests in this course.					
	5.91	6.71	-1.02	2.54	1.75
16. Availability of the instructor for student consultation.					
	7.55	6.71	1.38	.78	1.86
17. Instructor's demonstration of the subject being taught.					
	8.91	8.52	1.51	.29	.79
18. Clarity of presentation of the textbook.					
	7.00	6.62	.51	1.28	2.19
19. Adequacy of amount of lab work in learning information in this course.					
	7.09	7.62	-.61	2.61	2.01
20. Adequacy of amount of homework in learning information in this course.					
	6.82	6.00	.94	2.25	2.29

TABLE 3--Continued

21. Fairness of amount of weight placed on homework for grading purposes.					
	6.09	5.62	.87	1.62	1.29
22. Adequacy of the number of chances to bring up your grade.					
	6.64	6.19	.69	2.14	1.37
23. Ability of the course material to hold your interest.					
	6.73	6.57	.20	1.76	2.11
24. Instructor's consideration and courtesy toward students.					
	8.36	7.43	2.04*	.64	1.40
25. Probable overall rating of this course by the entire class.					
	6.91	6.10	1.54	.90	1.57
26. Understandability of language used by the instructor.					
	7.73	6.86	1.80	1.21	1.28
27. Instructor's encouragement to student learning.					
	7.36	6.48	1.64	1.07	1.56
28. General organization of classroom procedures.					
	7.64	7.24	.79	1.07	1.41
29. Rating of instructor when compared with other college teachers you have had.					
	7.55	7.14	.78	1.08	1.46
30. Adequacy of the teaching of the application of concepts and principles.					
	7.18	6.67	.87	1.34	1.64
31. Integration of homework with lectures.					
	6.82	6.57	.37	1.47	1.87

* significant at the .05 probability level

TABLE 3--Continued

32. Instructor's freedom from annoying mannerisms.					
	8.00	7.62	.62	1.48	1.68
33. Adequacy of instructor's outlining the goals of course and how developed.					
	7.27	7.14	.19	1.54	1.93
34. Do all assignments require the same amount of time?					
	1.45	1.48	.11	.50	.50
35. Textbook used as: background; repetition; text; busy-work.					
	2.00	1.71	.79	1.04	.88
36. What grade do you think you will make in this course?					
	1.91	1.90	.02	.51	.68
37. What grade do you think you deserve?					
	1.55	1.81	.97	.66	.73
38. How would you classify this course: formal; half formal and half informal; informal; too informal.					
	1.82	2.24	1.28	.57	.97
39. Would you like to take another course dealing with this subject?					
	1.27	1.19	.52	.45	.39
41. Overall impression of the CAI course.					
		5.24			2.51
42. Adequacy of CAI course as a suitable method for teaching ELASTIC.					
		5.43			2.42
43. Level of interest in ELASTIC fostered by CAI course.					
		5.33			2.34
44. Rating of CAI course section dealing with number conversions.					
		6.10			1.44
45. Rating of CAI course section dealing with ELASTIC1.					
		6.57			1.76

TABLE 3--Continued

46. Rating of CAI course section dealing with ELASTIC2.	5.71	2.14
47. Rating of CAI course section dealing with ELASTIC3.	5.67	1.96
48. Rating of CAI course section dealing with ELASTIC macros.	5.95	2.10
49. Rating of quizzes in the CAI course.	6.00	2.07
50. Rating of facilities in the CAI laboratory.	7.05	1.79
51. Rating of assistance given in the CAI laboratory.	6.95	1.68
52. Rating of general environment in the CAI lab.	6.67	1.94
53. Was it better for you personally to participate in a self-paced situation?	2.76	1.31
54. Would you take C.S. 310 again through CAI?	1.57	.90
55. Had you had any previous experience with CAI prior to C.S. 310?	1.86	.35
56. Did you finish the program assignments before the due date?	2.76	.92
57. Did you have much trouble scheduling time to take instruction from the CAI course?	2.76	.53

Discussion:

This experiment has shown a number of advantages and one possible disadvantage for using CAI to teach assembly language programming. In none of the hypotheses tested, did the CAI group perform significantly worse than the lecture group, in fact, both groups performed equally well in most instances. When there was a significant difference between the groups, the CAI group scored higher or performed more outstandingly.

The use of CAI reduced the total instruction time by more than half. From the data in the experiment it was shown that the total average time spent per student in the CAI group was 9.7 hours while the same amount of course material required 24 hours of formal classroom instruction. This saving of time is most significant in that it demonstrates the fact that when CAI is used, the instructor is available a greater amount of time for student consultation and personal attention.

The use of CAI tended to make the students more independent and confident. It was shown that the CAI group made fewer office visits and when they did consult with the instructor, the questions that they asked were relevant and fairly advanced in nature. The students in the lecture group who made office visits tended to make a greater number of visits with less relevant questions being asked than those asked by the students in the CAI group. The students in the lecture group tended to make office visits more for the purpose of getting acquainted with the instructor and vice-versa than actually getting solutions to their problems.

The use of CAI did not produce any significant difference in the student achievement produced by the course, when considering examination scores and program grades. There was really no reason to expect differences in means on tests or programs because usually in a credit-course in college, A students will do what is necessary on their own time to get an A, the same for B students, C students and so on Thus there is no way to measure the time spent reading manuals, talking with consultants and other students, etc. A true test of comparative efficiency would include these items.

It was noticed that the programs written by the students in the CAI group tended to be more "stereotyped" than those of the lecture group, as indicated by the smaller variance in the CAI group in the case of four of the programs. This may have been due to the absence of an adequate number of examples and alternate methods for solutions to given problems presented by the CAI course. The linear nature of the CAI course and the lack of a high degree of individualization might also help account for this finding. It may be possible to revise the CAI course so as to cause the students learning from this particular course to produce more original and imaginative programs. An important outcome of revisions based on this formative evaluation would be an increase in variance of program ratings at the high end by enabling more students to obtain outstanding program ratings.

The attitude of the CAI group did not differ significantly from the lecture group. There were students in the CAI group that seemed to rebel at the idea of being taught by a machine; however, the majority of the CAI group was most enthusiastic and had nothing but praise for the CAI method of instruction. The primary drawback for the CAI students was the inability of the computer to answer questions. Poor answer-processing was also a disadvantage. The course author and the coder had not used a number of the sophisticated COURSEWRITER functions because of inexperience in using the language and errors in system documentation in some instances. The student could; however, enter questions as responses to the CAI program. The instructor then obtained a listing of these questions periodically and answered the majority of these questions in class when the CAI group met. Class meetings for the CAI group were held once every two weeks explicitly for this purpose.

The results presented above seem to indicate that CAI is applicable and practical for teaching assembler language programming. The particular CAI course used in the experiment appears to be valid and useful when used in combination with an instructor as described herein, in that students who are competent and confident in assembly language programming are produced.

Course Revisions

A significant number of revisions and modifications to the instructional program and the environment in which it executed were suggested as a result of analysis of the student responses to questions asked in the course and from numerous other data collected throughout the formative evaluation. Major revisions made to the program included additional facilities for processing student responses, removal or clarification of ambiguous statements and questions, and the addition of curriculum concerning programming techniques and procedures. It was hoped that these changes and the provision for more on-line interaction with the CAI program would increase the variance of program quality.

As the situation stood at the conclusion of the formative evaluation, the student took instruction in programming from one computer (IBM 1440) and wrote his ELASTIC programs for execution on another computer (CDC 6600). It was determined that possibly more effective CAI instruction could be realized and a higher motivation toward the using of computers cultivated if the student were to receive instruction and be able to submit ELASTIC programs using the same terminal. To achieve this situation either a new ELASTIC assembler/interpreter had to be written for the IBM 1440 or the CAI course had to be implemented for use on the 6600. The latter course of action was chosen when the IBM 1440 computer was removed from the CAI Laboratory and the Laboratory embarked on a new project involving the use of an IBM 360/50 with COURSEWRITER III and APL.

An instructional system developed at Purdue University called PICLS was chosen to be used as an interim CAI facility on the CDC 6600. PICLS was revised for use on the 6600 during the Summer and Fall of 1969. During this same period of time, the CAI course to teach ELASTIC was translated from COURSEWRITER I into the PICLS author language. To accomplish the translation a program written in FORTRAN was prepared. The translator was required due to the large size of the CAI course (about 10,000 cards). Difficulties encountered in the translation process were attributable to such things as: the two languages having different character sets; the presence of upper and lower case letters in COURSEWRITER but only upper case letters in PICLS; a different set of system functions for analysis of student responses in each language; and different semantic interpretation of operation codes whose mnemonic designations were the same in both languages.

Once the CAI course was translated into PICLS, the curriculum revisions and additions were made on the basis of data collected during the formative evaluation. In particular, a great amount of effort was expended in reviewing the responses made by some 90 students who had taken the CAI course since its inception. This review resulted in the addition of a significant amount of program code designed to analyze student responses. Since the output terminal to be used on the 6600 was a teletypewriter, instructional material on the use of the teletype as well as the RESPOND time-sharing system (under which PICLS operates) was added to the instructional program.

When conversion and revision of the CAI program had been completed, it was discovered that a number of features described in the documentation of the PICLS system did not function properly or did not exist. This fact resulted in the addition of a new operation code (XO) to the PICLS author language and a number of extensive changes in the CAI course. The changes in the CAI course were accomplished primarily through the use of a program in SNOBOL written by T. W. Pratt. This program inserted the XO operation code at the appropriate places in the program code.

The final version of the instructional program was first run through PICLS in December 1969. Principal reasons for the course load-time exceeding eight minutes were the physical length of the course as well as character manipulation and storage routines within the PICLS system. In addition to this intolerably long load-time, the PICLS system could only accomodate one student taking the CAI course at a time. It was decided, therefore, to subdivide the CAI course into eighteen subcourses so that the entire course did not have to be loaded each time a section was used. Explanation on the course structure and instructions for sequencing through the subcourse had to be added to the instructional program so that students could use the course. Also added were facilities for linking the subcourses together.

Before this new version of the course could be tested, the problem of PICLS only being able to handle one student on the course at a time was solved. The solution entailed establishing the eighteen subcourses as public files on the RESPOND time-sharing system. Proceeding in this manner caused more changes to be made to the instructional program since new linking and sequencing procedures were necessitated. In an effort to save valuable computer storage, the CAI course was packed into a minimum amount of space before being built as public files. The CAI course became operational on the CDC 6600 on February 1, 1970. Chapter IV describes more fully the operational environment in which this new version of the instruction course resided.

CHAPTER IV

Operational Environment

The purpose of this chapter is to describe the environment in which the final version of the instructional program to teach the ELASTIC programming language executes, both from a hardware and software point of view. Also included in this chapter is a discussion of interfacing the ELASTIC language with the instructional system.

Hardware Configuration

The hardware configuration which the instructional program utilizes may be divided into two categories: the computer mainframe and the terminal and telecommunications equipment through which the student receives and transmits information. The computer mainframe is a Control Data 6600 computer. This computer consists of 11 processors, one large central processor and 10 peripheral processors. Each processor is an independent unit in its own right, however, the processors' actions are integrated to form a smooth-running system. The central processor contains extremely high-speed arithmetic and logic functions. The central processor possesses 131,072 words (60 bit) of magnetic core memory which is accessible to all 11 processors. Each peripheral processor possesses its own 4096 word (16 bit) memory. All communication between processors takes place in central memory. The central processor is a binary, sequential computer capable of performing 3 million instructions per second. Each word in central memory may contain from 2 to 4 computer instructions or a number with up to 17 decimal digits of significance.

The peripheral processors serve to control and schedule all communications between the computer and peripheral devices such as printers, readers, tape units, etc. In addition the peripheral processors control the sequence of operations performed by the central processor. Since all input/output is handled by the peripheral processors, the central processor is only concerned with performing high-speed arithmetic calculations ("number crunching"). A maximum of seven jobs or programs may reside in central memory at one time. A peripheral processor controls which program is being executed at any one instant so that a certain degree of functional concurrency is achieved in execution of programs by the central processor.

In order to illustrate the true magnitude of the 6600 system, a few of the peripheral equipments connected to the 6600 are listed below.

1. An operator console with typewriter input facility and two cathode-ray tube displays for monitoring of the system

2. Six magnetic tape drives rated to handle magnetic tape recording at 200, 556 and 800 bits per inch density. Information can be read or written at the rate of 120,000 characters per second.

3. Two card readers rated at 1200 cards per minute

4. Four line printers capable of printing 1000 lines per minute, with each line containing 136 characters.

5. One card punch rated at 250 cards per minute

6. Four magnetic disk storage units, each of which can store 84,000,000 characters of information with a disk access time at 30 to 100 milliseconds; four simultaneous accesses, one per disk, may be initiated at one time.

7. A microfilm recorder for recording graphical and printed data on 35mm film at the rate of 30 frames per second

8. A graphical display console with light pen and typewriter input for visual display of and user interaction with graphical displays

9. Calcomp 111 controller and 763 plotter for output of graphical plots

10. Two communication controllers which facilitate the linking of eight other computers with the 6600 so that the computational power of the 6600 may be more fully utilized

11. 500,000 words (60 bit) of extended core storage to permit faster, more efficient computer usage

12. Telecommunications equipment which permits remote access to the 6600 by up to 64 teletypes

The terminal at the remote site may consist of a Model ASR 35 teletypewriter, a Model ASR 33 teletypewriter, a CC30 CRT terminal, or a Data Point 3300 CRT terminal. Communication to and from the computer is accomplished over voice-grade telephone lines connected to a Model 101C Data Set or an acoustic coupler. Full duplex data transmission lines can handle information transmission at the rate of 2400 bits per second or 110 bits per second.

Software Configuration

As with the discussion of hardware configuration, the software configuration which is used by the instructional program will be described in two parts. It will be discussed first with respect to writing and compiling the instructional program and then relative to presenting a lesson to the student.

The CDC 6600 computer system at the University of Texas at Austin is operated under the control of the UT1 Operating System. The PICLS instructional system is a program written in FORTRAN and COMPASS that executes as a normal job under the UT1 operating system.

PICLS is composed of a number of subsystems which are as follows: an INPUT/EDIT system for building and modifying a course, a PRINT system for listing all or parts of a course, a COMPILE system for translating a course into executable form, a system for manipulating files, and an interpreter or LESSON system for presenting the course material to the student and processing his responses. The course author programs a course by writing statements in the PICLS author language. These statements are logically grouped into sections and the sections are grouped into a course or courses. Appendix E presents a more detailed description of both the PICLS author language and the commands which call the various PICLS subsystems.

Once the author has coded the instructional program, the program is compiled (using the COMPILE subsystem). The program to be compiled can be submitted as data to the COMPILE system either in "batch" mode where each statement in the program is punched on a card or in "teletype" mode. In teletype mode, the author keys each statement individually into the teletype. In either case, the program is compiled and recompiled until all syntax errors are removed at which time the course can be tested through use of the LESSON subsystem of PICLS. The LESSON system causes a compiled section to be executed, that is, presented to the student via the teletype.

When PICLS operates in teletype mode, the time-sharing system available on the CDC 6600 is used. The time-sharing system is called RESPOND and is briefly described in Appendix F. In teletype mode PICLS executes as a "conversational" program, that is, the user sits at a teletype terminal and "converses" or interacts with the PICLS system as it is running. This conversational facility is obtained by using the CONVERSE feature of the RESPOND system. Normally, unless the course to be compiled is very large as is the ELASTIC instructional program, it is entered in teletype mode. So then an author can sit down at a teletype and both create and take a course using teletype mode as a normal course of action.

After the ELASTIC course was compiled in batch mode, the eighteen subcourses as described in Chapter II were stored as public files so that they were accessible to anyone using the RESPOND time-sharing system. An additional reason for storing the courses in this way was that otherwise only one student could take the course at a time. When the course is not being used regularly as it is now, it is stored on magnetic tape and read into 6600 system COMMON files each time the course is desired.

A student receives instruction from a subcourse by logging onto the RESPOND system and then submitting a conversational job which calls for execution of the PICLS program using a particular subcourse that the student wishes to take. Instruction from a section of the subcourse is obtained by using the LESSON subsystem of PICLS and giving the name of the section desired.

In summary, the teletype mode of PICLS is achieved by executing the PICLS program as a conversational job within the RESPOND time-sharing system. The RESPOND system in turn is a program being executed as a job under the UT1 operating system.

ELASTIC Language Interface

The last section of this chapter is concerned with briefly explaining the interface between the ELASTIC programming language and the CAI course to teach the language. At first glance (and even at second glance) the problem of providing facilities for instruction and execution of the same language during one session on the teletype seemed insurmountable.

However, what sometimes seems to be the most difficult problem turns out to have the simplest solution.

Since PICLS executes using the CONVERSE feature of RESPOND and since RESPOND is used primarily to create files (programs and data) and to submit these files to the 6600 for execution, the problem of interfacing ELASTIC language facilities with the instructional course took the form of letting the user switch between the conversational and nonconversational modes of RESPOND. To achieve this ability to switch between modes, the student was required to know more about using RESPOND than he would have otherwise known. This additional instruction on RESPOND was accomplished quite easily, however, through the use of a handout entitled "ELASTIC on RESPOND" that appears in Appendix B, and a two-hour lecture on the fundamentals of file manipulation using RESPOND. The solving of the interface problem would not have been as straightforward had PICLS executed as a "stand-alone" system or had the ELASTIC system not been accessible from RESPOND. It should be noted that RESPOND can only access systems that require less than 100,000 (octal) words of memory and that ELASTIC required 105,000 (octal) words of memory until it was recompiled by a new and more efficient FORTRAN compiler (about a year ago) which reduced its required field length or memory requirement to 76,000 (octal) words.

To summarize the interface facility then, the student logs onto RESPOND and then logs onto PICLS to receive instruction from a subcourse. At anytime during presentation of curriculum in the subcourse, the student can exit from PICLS and return to the nonconversational mode of RESPOND. In this mode the student creates a file consisting of ELASTIC language statements. He then submits the ELASTIC program for execution, receives and examines the output produced by the program and determines his next action. When he has received satisfactory results, the student re-enters the PICLS system to receive instruction from the CAI course once again. A sample teletype session in which both the instruction and execution modes are illustrated appears in Appendix I of this paper.

CHAPTER V

Summative Evaluation

Introduction

The purpose of this chapter is to describe the second experiment that was conducted as an evaluation of the computer-assisted instruction course to teach the ELASTIC programming language. This experiment was performed during the Spring of 1970 at the University of Texas at Austin. As in the formative evaluation, the primary purpose of the experiment was to investigate the practicality of employing different mixes of CAI techniques and batch or conversational job submission in teaching assembly language programming.

The same qualifications regarding a "horse race" between CAI and non-CAI versions of a course as were stated in Chapter III apply to this study. While having some of the forms and some of the terminology of behavioral science research and while being described as a summative evaluation, the most important long-range outcomes of this study, it was hoped, would be information to guide the development of new and more powerful CAI software and hardware systems to teach programming.

Hypotheses

The following hypotheses were tested in this experiment.

1. The CAI group will complete course instruction significantly faster than the lecture group;

2. The CAI group will make significantly fewer personal visits to the instructor;

3. The CAI group will ask significantly more advanced questions of the instructor in office visits;

4. There will be no significant difference between the CAI and lecture groups with respect to mean score on examinations;

5. There will be no significant difference between the CAI and lecture group with respect to mean grade on computer programs written;

6. There will be no significant difference in variances in programs written by the CAI and lecture groups.

Results from the formative study caused numerous hypotheses to be changed in the summative evaluation. Instead of testing a hypothesis that the CAI group will score higher on quizzes and computer programs written, the revised hypotheses were that both groups would score about the same on quizzes and programs, since as was stated previously, an A student will usually do enough on his own to get an A, a B student a B, and so on, regardless of the instructional treatment. An additional hypothesis regarding variance in programs written by the different groups was included as a result of findings in the formative evaluation. The addition of more examples and provisions for on-line execution of programs during an instructional session should tend to offer the opportunity for more practice and hence more variety in programs written by students in the CAI group.

From a study of results relative to attitude in the formative study, the attitude scale used in that study was determined to have only limited usefulness. Two new instruments to measure attitude which have been used in other studies were administered as an exploratory effort. The results obtained from these instruments will be presented in the discussion section in this chapter. No hypothesis dealing with attitudes was stated in this study because of earlier findings and because of the use of new instruments in this study.

Method

The method employed in this experiment was very similar to that utilized in the formative evaluation discussed in Chapter III. The subjects for the experiment were students enrolled in sections 2 and 8 of Computer Sciences 310 at the University of Texas at Austin. One class served as the control group (lecture group) while the other class served as the CAI group. Due to the availability of very limited facilities with which to perform this evaluation, it was necessary to limit the number of students in each group to ten. Classification and the grade made in the prerequisite course, Computer Sciences 404G, were used as the criteria for determining which students who registered for sections 2 and 8 during the Spring registration would be allowed to stay in those sections and participate in the experiment. All students not chosen were requested to change to other sections of the course. The criteria mentioned above were employed in an effort to insure that the groups had approximately the same entering abilities, as far as could be determined from classification and grades. Since the students were not randomly placed in groups and due to the extremely small number of subjects involved in this study, constraints on the interpretation of results reported here should be recognized.

The lecture group received instruction through formal classroom lectures while the CAI group received their instruction through the revised computer-assisted program described in Chapters II and III. The CAI course was administered through the use of two Model ASR35 teletypewriters which were driven by the PICLS instructional system as it was executed in conversational mode on the CDC 6600 computer system at the University of Texas at

Austin. The reasons for using the CDC 6600 system and a description of the modifications and revisions made to the CAI course prior to conducting this experiment are presented in the last section of Chapter III. As in the formative study the primary objective of the CAI program was to teach the ELASTIC programming language.

At the conclusion of the course, a comparison was made between the groups according to the following criteria: examination grades, program grades and variance in program grades, responses to attitude questionnaires, number of office visits made and ratings of conversations between the students and the classroom instructor.

The instructor for both sections was Fred Homeyer. Both classes were given formal lectures on general computer concepts during the first three weeks of the semester. Each class was then given an hour examination in order to determine the statistical procedures to be employed for analysis of subsequent data to be collected. After this test, the mean score of each class indicated that analysis of variance techniques could be used to analyze the data collected in the experiment. That is, no difference, so students were assumed to be equally capable, precluding necessity for covariance analysis.

After the first quiz, the instructional treatment given to the two classes varied greatly as the test group or CAI group took their instruction in ELASTIC using the CAI program. Each student participated in a self-paced environment in that he scheduled CAI sessions at times convenient to him. The CAI group met in the classroom for two hours prior to each of the remaining two quizzes so that any difficulties they had experienced could be rectified. The lecture group for this study received instruction in ELASTIC through classroom lectures. Both classes were given two hour examinations and a final examination over the ELASTIC language and its use. These quizzes may be referenced in Appendix C of this paper. Each student in both groups wrote and tested a total of eight computer programs. These programs are also illustrated in Appendix C.

The experimental design used for the summative evaluation was somewhat different from that employed in the formative study. Figure 5 in Chapter III can still be used to illustrate the dependent variables considered in the overall experiment design; however, some changes were made in the mode in which students developed and tested their computer programs. Both the lecture and CAI groups were subdivided into a "batch" group and a "teletype" group. The students in the batch group wrote and submitted their programs for execution in batch mode while the students in the teletype group developed their programs and submitted them for execution using the RESPOND time-sharing system via two teletypewriters. The modified design is illustrated in Figure 6.

	Batch Group	Teletype Group
Lecture Group		
CAI Group		

Figure 6

Summative Evaluation Design

Both the lecture and CAI groups consisted of ten students each. The batch and teletype subgroups of each group contained five students each. The total number of students participating in the experiment was relatively small due to the fact that only two teletypewriters were available for use in the study. The time that the RESPOND system was operational, scheduling problems and anticipated CDC 6600 system problems were also factors which limited the number of participants.

To test the first hypothesis, the average time spent by a student taking the CAI course was compared against the time spent by each student in the lecture group sitting in formal classroom lectures. A log of instruction time was kept by each student throughout the semester. In addition a summary of total computer time, RESPOND terminal time, number of jobs run and number of output pages produced by each student were obtained through facilities provided by the University of Texas Computation Center.

To test the second hypothesis, the average number of visits per student in both groups was computed and these figures were compared and tested for significant difference of means using a double classification analysis of variance. A diary of visits for each student was kept throughout the semester by the instructor.

To test the third hypothesis, the questions asked by the students in personal visits with the instructor were rated according to the scale: elementary, intermediate and advanced. The average rating for each group was computed. As in the formative evaluation, the results of this test should be considered in the perspective that prejudices and bias may have entered into the results, however, objectivity on the part of the instructor was strived for at all times.

The mean scores on examinations taken by both groups were used to test the fourth hypothesis. A double classification analysis of variance procedure was used to determine the presence of significant differences between groups.

To test the fifth hypothesis the mean grade on each of the eight programs was computed for both groups and analysis of variance techniques were employed to determine significant difference. The same criteria for grading program assignments that was employed in the formative study was used here. These criteria were described in Chapter III. A student assistant named Paul Pedigo graded the programs in this study while Mike Cunningham graded the programs written in the formative evaluation. Mike Cunningham was not available to act as grader for the summative study.

To test the sixth hypothesis, variance of programs written by the CAI group and the lecture group, a variance ratio F-test was performed on each of the eight programs. Only the CAI-batch and lecture-batch groups were considered in testing this hypothesis in order to compare the findings from this study with the results from the formative evaluation.

Concerning a study of student attitudes, all four subgroups completed a Semantic Differential rating of the concept "computer" and the concept "automobile" using a number of bipolar adjective scales that were reportedly used by Mathis, Smith, and Hansen [21]. The rationale for the Semantic Differential methodology is described in Osgood, Suci and Tannenbaum [23]. The more positive adjective of each pair was assigned a value of 7 and the less positive adjective was assigned a value of 1.

The CAI group also completed a form of the instrument developed by Brown [6] used to measure student attitudes toward computer-assisted instruction. The particular form of the questionnaire used in this experiment was reported in [21]. The students circled one of the five responses for each statement. The responses ranged from strongly agree to strongly disagree on most questions. The responses for questions requesting the frequency of some occurrence ranged from very often to very seldom. An internal consistency reliability coefficient of .89 has been reported for the original form of the Brown scale as reported in [6]. Mathis, Smith and Hansen [21] reported that the versions of the Brown scale they used had a Kuder-Richardson Formula 20 reliability of .82 for 158 Florida State undergraduates. The version of the Brown scale used in this study was the same as that used by Mathis, Smith and Hansen.

A teletype questionnaire developed by the author was given to the teletype subgroups of both groups. In this questionnaire the student circled the one response that most nearly reflected his opinion toward the statement in question. The results of the student responses to these instruments were analyzed and tested for significant difference between groups. Refer to Appendix H where copies of these instruments appear.

The results of the experiment appear in the following section.

Results

The results of the experiment will be presented in order by the hypotheses as listed in the second section of this chapter. Before presenting the results, the "profiles" for the students who participated in the study are proffered in Tables 4, 5, 6, and 7. Table 4 contains the profiles for students in the lecture-batch group and Table 5 contains the lecture-teletype group profiles. The CAI-batch group profiles appear in Table 6 while Table 7 presents the profiles for students in the CAI-teletype group. The profile gives the following information for each student: student number, sex, classification, major, previous experience with CAI, number of previous computer courses taken, grades in previous courses, languages in which the student can program, number of office visits made during the semester, average rating of questions asked by the

TABLE 4

STUDENT PROFILES IN THE LECTURE-BATCH GROUP

Student Number	1	2	3	4	5
Sex	F	F	M	M	M
Classification	Jr.	Soph.	Fr.	Jr.	Sr.
Major	Physics	Math	Acct.	Math	Pre-Med
Previous CAI	No	Yes	No	No	No
Previous C.S.	1	1	1	1	1
Grades in previous C.S. course	B	A	C	B	B
Language Used *	E & F	Algol	E & F	E & F	E & F
No. Office Visits	0	4	3	3	1
Question Rating	-	1.75	1.0	2.0	3.0
Program Average	91.3	93.3	71.0	97.4	90.7
Quiz 1	79	76	77	78	84
Quiz 2	87	80	63	77	99
Quiz 3	85	96	94	95	90
Final Exam	90.4	91.2	87.2	88.0	90.4
Total Computer Time	.0683	.0606	.0749	.0307	.0290
No. Jobs Run	57	48	59	23	25
Total RESPOND Time	-	-	-	-	-

* E & F denotes ELASTIC and FORTRAN

TABLE 5

STUDENT PROFILES IN THE LECTURE-TELETYPE GROUP

Student Number	1	2	3	4	5
Sex	F	F	M	F	M
Classification	Gr.	Jr.	Soph.	Jr.	Sr.
Major	Media Educ.	Psy.	E.E.	Math	Math
Previous CAI	No	No	No	No	No
Previous C.S.	1	1	2	1	1
Grades in previous C.S. course	C	C	B & A	A	A
Language Used *	E & F	E & F	Algol Ftn	E & F	E & F
No. Office Visits	6	3	5	0	2
Question Rating	1.68	1.33	2.8	-	2.0
Program Average	94.8	93	96.2	92.3	94.4
Quiz 1	82	72	89	86	81
Quiz 2	60	67	98	94	73
Quiz 3	76	77	93	88	91
Final Exam	88	91.2	93.6	94.4	88.8
Total Computer Time	.1040	.0387	.0500	.0961	.0630
No. Jobs Run	237	120	135	207	142
Total RESPOND Time	21.511	17.874	12.926	21.384	13.847

* E & F denotes ELASTIC and FORTRAN

TABLE 6

STUDENT PROFILES IN THE CAI-BATCH GROUP

Student Number	1	2	3	4	5
Sex	M	M	F	M	M
Classification	Jr.	Soph.	Fr.	Gr.	Fr.
Major	Math	Hist.	Math	Math	Math
Previous CAI	No	Yes	No	No	No
Previous C.S.	1	1	1	1	1
Grades in previous C.S. course	A	B	A	A	?
Languages Used *	E & F	E & F	E & F	E & F	E & F
No. Office Visits	2	1	5	2	1
Question Rating	2.0	2.0	2.0	1.5	2.0
Program Average	94.1	89.6	99	91.4	71.4
Quiz 1	83	78	87	88	81
Quiz 2	78	86	84	84	99
Quiz 3	99	91	89	82	88
Final Exam	89.6	91.2	91.2	92.8	88.8
Total Computer Time	.5792	.4890	.3649	.6703	.3383
No. Jobs Run	149	163	169	246	111
Total RESPOND Time	13.503	12.945	12.699	15.218	14.392

* E & F denotes ELASTIC and FORTRAN

TABLE 7

STUDENT PROFILES IN THE CAI-TELETYPE GROUP

Student Number	1	2	3	4	5
Sex	M	M	M	M	M
Classification	Spec.	Jr.	Gr.	Jr.	Fr.
Major	Math	Math	Stat.	Math	Undet.
Previous CAI	Yes	No	No	Yes	Yes
Previous C.S.	1	1	2	1	1
Grades in previous C.S. courses	B	A	A & A	B	A
Languages Used *	E & F	E & F	E & F	E & F	E & F
No. Office Visits	2	2	4	3	3
Question Rating	1.0	3.0	2.0	2.0	3.0
Program Average	79.1	97.1	95.5	92.6	99.3
Quiz 1	81	84	64	83	81
Quiz 2	58	92	88	96	93
Quiz 3	64	99	94	76	95
Final Exam	65.6	96.8	87.2	90.4	95.7
Total Computer Time	.3230	.5032	.4291	.4053	.7878
No. Jobs Run	178	298	286	252	574
Total RESPOND Time	21.385	38.428	43.683	30.080	57.477

* E & F denotes ELASTIC and FORTRAN

student (1 = elementary, 2 = intermediate and 3 = advanced), the average of the grades made on programs, the grade made on each examination and on the final examination, total computer time used, number of jobs run and total RESPOND terminal time.

In testing the hypotheses, a double-classification analysis of variance to test for significant difference between mean was used when all four subgroups were considered while a single-classification analysis of variance was used when only two subgroups were considered, for example, the CAI attitude questionnaire given only to the CAI group.

Hypothesis (1): The CAI group will complete course instruction significantly faster than the lecture group.

Results:

	Lecture Group	CAI Group
Number of students	10	10
Average time/student	24 hours	13.75
S.D.	0.0	1.04

Conclusion: Hypothesis accepted; the CAI group completed course instruction almost twice as fast.

Hypothesis (2): The CAI group will make significantly fewer personal visits to the instructor.

Results:

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Average number of visits/student	2.2	3.2	2.2	2.8
S.D.	1.64	2.38	1.64	.83

Analysis of Variance Source	Mean Square	F	Prob
Lecture vs. CAI	.2	.06	.79
Batch vs. TTY	3.2	1.08	.31
Interaction	.2	.06	.79

Conclusion: Hypothesis rejected; the CAI group did not have significantly fewer office visits per student. There was no significant difference in number of office visits at the .05 probability level.

Hypothesis (3): The CAI group will ask significantly more advanced questions of the instructor in office visits.

Results: (The questions were rated as follows: 1 = elementary, 2 = intermediate, 3 = advanced)

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Average Question rating	1.55	1.56	1.9	2.2
S.D.	1.12	1.02	.22	.83

Conclusion: Evidence offered as a test of this hypothesis is represented by the mean above. No statistical test was performed on this data because of small group size and the subjective rating scale used by the experimenter. The reader may examine the raw data in Tables 4, 5, 6 and 7. It can be seen that the difference in question quality found in the formative study was not replicated. Possible reasons are advanced in the discussion section.

Hypothesis (4): There will be no significant difference between the CAI and lecture groups with respect to mean score on examinations.

Results:

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Quiz #1:				
Average	78.8	82.0	83.4	78.6
S.D.	3.11	7.44	4.15	8.26

Analysis of Variance Source	Mean Square	F	Prob
Lecture vs. CAI	1.8	.052	.815
Batch vs. TTY	3.2	.093	.760
Interaction	80.0	2.33	.142

Conclusion: No significant difference between groups on Quiz #1 at the 0.05 level of significance.

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Quiz #2:				
Average	81.2	78.4	86.2	85.4
S.D.	13.23	16.77	7.75	15.58

Analysis of Variance Source	Mean Square	F	Prob
Lecture vs. CAI	180.0	.948	.653
Batch vs. TTY	16.2	.085	.770
Interaction	5.0	.026	.867

Conclusion: No significant difference between groups or subgroups on Quiz #2 at the 0.05 level of significance.

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Quiz #3:				
Average	92.0	85.0	89.8	85.6
S.D.	4.52	7.96	6.14	14.97

Analysis of Variance Source	Mean Square	F	Prob
Lecture vs. CAI	3.2	.037	.843
Batch vs. TTY	156.8	1.81	.194
Interaction	9.8	.113	.739

Conclusion: No significant difference between groups or subgroups on Quiz #3 at the 0.05 level of significance.

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Final Exam:				
Average	88.44	91.2	90.72	87.14
S.D.	1.73	7.82	1.55	12.66

Analysis of Variance Source	Mean Square	F	Prob
Lecture vs. CAI	9.66	.222	.647
Batch vs. TTY	4.14	.095	.758
Interaction	35.64	.820	.618

Conclusion: No significant difference between groups or subgroups on the final examination at the 0.05 level of significance.

Conclusion on hypothesis: Accept the hypothesis; there was no significant difference between the CAI and lecture groups with respect to mean score on examinations.

Hypothesis (5): There will be no significant difference between the CAI and lecture groups with respect to mean grade on computer programs written.

Results:

Rather than presenting each program separately, only the combined average grade on all programs will be considered. (A study of each program considered separately indicated no significant difference between subgroups or groups at the 0.05 level of significance.)

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Average grade on all programs	88.74	94.14	89.1	92.7
S.D.	10.25	1.53	10.51	7.99

Analysis of Variance Source	Mean Square	F	Prob
Lecture vs. CAI	1.40	.019	.884
Batch vs. TTY	101.7	1.44	.246
Interaction	3.96	.056	.810

Conclusion: Hypothesis accepted; both groups did equally well with respect to grade on programs written.

Hypothesis (6): There will be no significant difference in the variance of programs written by the CAI and lecture groups.

Results:

The results of this study appear in Table 8. A variance ratio F-test was used to determine variance in programs written by the CAI-batch and the lecture-batch groups. In Table 8 the analysis for all four subgroups is included for completeness.

TABLE 8

DATA FOR PROGRAMS WRITTEN IN SUMMATIVE STUDY

Program	Lecture-Batch		CAI-Batch		F	Lecture-TTY		CAI-TTY		F
	Mean	S.D.	Mean	S.D.		Mean	S.D.	Mean	S.D.	
1	86.4	13.2	91.8	5.9	5.00	92.8	1.9	89.8	11.4	36.0*
2	89.2	12.1	90.6	5.4	5.02	89.4	9.2	91.0	8.3	1.23
3	89.8	3.5	92.4	6.5	3.44	93.0	7.9	95.4	5.8	1.85
4	88.4	4.2	76.6	43.0	104.*	91.6	10.0	96.0	4.1	5.94*
5	92.0	6.0	90.8	8.3	1.91	95.0	3.3	93.6	6.1	3.41
7	94.8	3.5	86.4	19.2	30.*	96.2	2.4	95.8	4.6	3.67
10	92.8	5.7	93.2	5.8	1.03	92.0	2.1	91.2	8.1	14.8*
11	76.8	43.0	91.0	8.6	25.0*	98.0	1.7	88.4	17.8	109.*
Average	88.7	10.2	89.1	10.5	1.05	94.1	1.5	92.7	7.9	27.7*

* Significant at the .05 probability level.

<u>Conclusion</u>: Reject the hypothesis. Programs 4, 7 and 11 showed significantly different variance between the two groups considered and programs 1 and 2 are very near to being significantly different between the two groups at the .05 level of significance. On programs 4 and 7 the lecture-batch was less varied while on program 11 the CAI-batch group showed less variation.

<u>Discussion</u>

The results of this experiment can be summarized as follows: the CAI group completed the course instruction twice as fast as the lecture group, considering only time spent in the PICLS ELASTIC course itself; there was no significant difference between the lecture group and CAI group with respect to number of office visits; there was no significant difference between the groups with respect to average grade of programs written, quiz grades or the final examination; the variances on programs written by the CAI-batch and the lecture-batch groups were significantly different in three cases.

It should be noted that care should be taken in comparing the formative and summative evaluations presented in this paper due to the fact that curriculum changes in the Computer Sciences department resulted in students with different backgrounds participating in the two experiments. Students participating in the formative study had no prior experience with ELASTIC while most students in the summative study had received instruction in ELASTIC1 in the introductory computer sciences course. It was found that very few students remembered or understood anything about ELASTIC upon their beginning Computer Sciences 310, however. In addition, when considering the variance of programs in hypothesis 6, it should be noted that a different person graded the programs in the two studies. Therefore, comparing results becomes quite difficult in this case. It may be noted that the average time to take the CAI course increased from 9.7 hours in the formative study to 13.75 hours per student in the summative evaluation. There may have been a number of reasons for this increase including slower output rate of the teletypewriter as compared to the IBM 1050 typewriter terminal, the additional amount of text and questions incorporated into the revised version of the CAI program and the facilities for on-line execution of small ELASTIC programs being included in the instructional situation. Also since the average time for student completion of the CAI course was obtained by averaging the total RESPOND terminal time used by each student in the CAI-batch subgroup, there is no assurance that some time at the terminal was not spent testing program assignments instead of receiving instruction from the CAI program. It should be noted; however, that according to the time logs kept by all students in the CAI group (both batch and teletype subgroups), the estimated time for course completion ranged between 12 and 15 hours.

A number of additional results concerning time used by students was obtained as a consequence of considering the first hypothesis tested in this experiment. These results will be discussed below. The average computer time used by each subgroup is presented in Table 9. The students in the CAI group used significantly more computer time than the lecture group. The additional motivation of having the "computer at your fingertips" in a time-sharing environment may have caused this result. Due primarily to the characteristics and speed of the CDC 6600 computer system, the average central processor time required to take the ELASTIC CAI course was approximately .4883 hours per student. At a cost of $200 per central processor hour, the CAI course cost per student was approximately $97.66 or &7.10 per student terminal hour.

TABLE 9

AVERAGE COMPUTER TIME USED

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Average computer time used (hrs/student)	.0527	.0704	.4883	.4897
S.D.	.0215	.0286	.1406	.1786

Analysis of Variance

Source	Mean Square	F	Prob
Lecture vs. CAI	.914	69.0	.000
Batch vs. TTY	.000	.034	.849
Interaction	.000	.025	.870

Another interesting result obtained from the summary of computer statistics furnished by the Computation Center concerns the number of jobs run per student. Table 10 presents the average number of jobs for each of the four subgroups. It should be noted when examining this table that taking instruction from the CAI course required submitting numerous jobs in conversational mode in order to execute the PICLS instructional system.

TABLE 10

AVERAGE NUMBER OF JOBS RUN

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Average number of jobs run	42.4	169.2	167.6	317.6
S.D.	17.3	50.27	49.29	150.76

Analysis of Variance

Source	Mean Square	F	Prob
Lecture vs. CAI	93571.2	13.37	.0024
Batch vs. TTY	95772.8	13.68	.0022
Interaction	672.8	.096	.7579

Due possibly to the motivation factor mentioned previously we see that there was a significant difference in number of jobs run by the lecture and the CAI groups and that there was a significant difference in number of jobs run between the batch and the teletype groups. There was no significant interaction between groups at the .05 level. While the CAI-batch subgroup ran an average of 167.6 jobs there is no way to determine which jobs were submitted in batch mode and which in teletype mode. It is interesting to note, however, the much greater number of average jobs submitted by the CAI-teletype subgroup. Although some of these jobs may have been of a housekeeping nature, that is, copying output to printer, copying data to cards, etc., and although the standard deviation for this subgroup is 150.76, the fact still remains that some students may have been more highly motivated when placed in the instructional environment in which all contact with the computer is through a time-sharing system.

Consideration of the total time spent at the teletype terminal by each subgroup also reveals some interesting results. Table 11 presents the average terminal time spent by students in each group.

TABLE 11

AVERAGE TIME SPENT AT TERMINALS

	Lecture-Batch	Lecture-TTY	CAI-Batch	CAI-TTY
Average time spent at terminal (hrs/student)	0.0	17.50	13.75	38.21
S.D.	-	4.04	1.04	13.69

Analysis of Variance

Source	Mean Square	F	Prob
Lecture vs. CAI	1483.85	28.93	.0002
Batch vs. TTY	2201.63	42.93	.0000
Interaction	60.38	1.17	.2942

From Table 11 it can be determined that there was a significant difference in amount of terminal time spent by students in the lecture and CAI groups and also a significant difference in amount of time spent by the batch and teletype groups. Even though the results above were expected, perhaps the most surprising result from this portion of the study is that the CAI-teletype subgroup spent on the average 21 more hours at the terminal than any other subgroup and spent an average of 7 more hours running programs than any other subgroup. It can also be noted that the teletype subgroups may have been more highly motivated to use the computer as a result of their respective instructional situations.

It should be noted that the results of testing hypothesis 2, the number of office visits per student, was not replicated in the summative evaluation. There may have been a number of reasons for this including the fact that the instructor's office was moved to a distant part of campus and that office hours were held in a room adjacent to the room housing the teletypewriters used by the CAI group. This made access to the instructor relatively more easy for students in the CAI group. Even though the results of the summative study do not repeat the finding of the formative study, it is felt that the results of the formative study may still be valid with respect to this hypothesis; however, a final conclusion awaits further testing.

As with hypothesis 2, the summative evaluation rejected hypothesis 3 concerning quality of questions asked by students in office visits. Here again, the factors mentioned above and the objectivity of the instructor may have caused the results to be different from those in the formative evaluation.

As expected there was no significant difference between the lecture and CAI groups with respect to examination scores and program grades. This finding has been made in numerous studies which compared the use of CAI techniques to formal lectures.

The CAI course may have achieved partial success in getting students in the CAI group to write programs which are not so "stereotyped." As in the formative study there were three cases in which programs written by the CAI and lecture groups varied significantly at the .05 level. In this case, however, the difference was observed on programs 4, 7 and 11 while in the formative evaluation the difference was in programs 2, 10 and 11. In two of the three cases where significant difference was found in the summative study the CAI group was more varied, thus indicating possibly that the CAI course in its revised form can cause students to write more original and imaginative programs. Although not considered in testing hypothesis 6, a study of the lecture-teletype and CAI-teletype subgroups relative to variance in programs written reveals that programs 1, 4, 7, 10 and 11 varied significantly between groups at the .05 level of significance. The lecture-teletype group varied more than the CAI-teletype group on program 4 while the CAI-teletype group wrote programs more varied on the remaining three programs mentioned above. This result provides some evidence to the claim of success of the CAI course in its revised form to foster more original and imaginative program design on the part of the students than did the original version of the course. Caution should be taken in studying these results, however, because as mentioned previously, a different person graded the programs in the summative study because the person who graded them in the formative evaluation was not available. Also the very small number of students in the subgroups makes variance estimates less reliable.

As was stated previously, no hypothesis concerning attitude was included in this study. However, two exploratory instruments which have been used to measure attitude in other studies were administered. The results from these appear below.

A discussion of the results of the Semantic Differential rating of "computer" appears below. Table 12 presents the relevant data while Table 13 presents the results from an analysis of variance on the seven cases that reflected significant difference in means of groups at the .05 level of significance.

TABLE 12

SUMMARY OF RESPONSES TO SEMANTIC DIFFERENTIAL

Adjectives	Lecture-Batch		Lecture-TTY		CAI-Batch		CAI-TTY	
	Mean	S.D.	Mean	S.D.	Mean	S.D.	Mean	S.D.
fast-slow	6.4	.89	6.8	.44	6.8	.44	7.0	.0
interesting-dull*	6.0	.70	6.8	.44	6.6	.54	6.8	.44
relaxed-tense*	4.2	1.3	4.0	1.5	1.8	.44	3.0	1.8
good-bad*	5.4	1.1	6.2	.44	6.6	.54	6.4	.54
fair-unfair	5.4	1.5	5.2	1.3	5.6	1.9	6.2	1.3
deep-shallow	5.8	1.3	5.4	1.1	5.2	1.6	5.8	1.3
valuable-worthless	6.4	.89	7.0	0.0	6.0	1.2	6.8	.44
active-passive*	5.2	1.3	6.2	.44	5.6	2.0	7.0	0.0
easy-difficult	3.8	1.9	3.6	1.3	2.4	1.1	3.0	1.8
flexible-inflexible*	3.8	1.3	6.0	.70	4.6	2.6	5.8	1.3
exciting-boring*	5.4	.54	6.8	.44	6.0	1.0	6.6	.54
pleasant-unpleasant	5.2	.83	6.2	1.0	5.2	1.7	5.2	1.0
encouraging-depressing	4.4	1.6	5.4	1.5	3.8	2.1	5.4	1.5
safe-dangerous	5.8	1.3	5.6	1.6	5.8	1.6	5.8	1.6
strong-weak*	5.4	1.3	6.2	1.3	4.2	.44	6.4	.54

* significant difference at .05 probability level

TABLE 13

DATA FOR SIGNIFICANTLY DIFFERENT ITEMS ON SEMANTIC DIFFERENTIAL

Adjectives	Analysis of Variance Source	Mean Square	F	Prob
interesting-dull	Lecture vs. CAI	.45	1.5	.237
	Batch vs. TTY	1.25	4.16	.055
	Interaction	.45	1.5	.237
relaxed-tense	Lecture vs. CAI	14.45	7.31	.015
	Batch vs. TTY	1.25	.632	.556
	Interaction	2.45	1.24	.281
good-bad	Lecture vs. CAI	2.45	4.66	.044
	Batch vs. TTY	.45	.857	.628
	Interaction	1.25	2.38	.139
active-passive	Lecture vs. CAI	.45	.290	.603
	Batch vs. TTY	11.25	7.25	.0153
	Interaction	.05	.032	.853
flexible-inflexible	Lecture vs. CAI	.45	.168	.689
	Batch vs. TTY	14.45	5.4	.031
	Interaction	1.25	.467	.510
exciting-boring	Lecture vs. CAI	.2	.444	.520
	Batch vs. TTY	5.0	11.11	.004
	Interaction	.8	1.77	.198
strong-weak	Lecture vs. CAI	1.25	1.25	.279
	Batch vs. TTY	11.25	11.25	.004
	Interaction	2.45	2.45	.134

Of the fifteen sets of adjectives, seven sets demonstrated significant differences between groups at the .05 level of significance. There were four instances of responses averaging less than the neutral point of 3.5. All four of these occurred in responses from the CAI group, in that they felt "computer" was more tense than relaxed and more difficult than easy. These results may have been due to the newness of procedures and facilities made available to the students in the CAI group.

The teletype groups felt "computer" was significantly more interesting, active, flexible, exciting and strong than did the batch groups. These results may have been due to the additional facilities that were provided to the teletype groups for program submission. These facilities seem to help remove the "black-box" notion of "computer" for the teletype groups as they appeared to be more aware of the potentials of the computer.

The lecture group felt "computer" was more relaxed than did the CAI groups. This result may have been caused by the absence of any new instructional situation provided for the lecture-batch subgroup or possibly because of the newness of techniques and procedures used by the CAI groups.

The CAI groups rated "computer" significantly better than the lecture groups when asked to consider the adjectives good and bad. This may have been caused by students in the CAI group realizing more of the potentials of using the computer as a result of the instructional environment in which they were placed.

Table 14 presents the results of the Brown scale used for measuring students' attitudes toward computer-assisted instruction. This instrument was given only to the CAI subgroups. The direction scored positive to CAI may be determined by reference to Appendix H of this paper.

TABLE 14

RESULTS OF THE BROWN SCALE FOR ATTITUDE TOWARD CAI

Item	Mean Ratings CAI-Batch	CAI-TTY
1. While taking CAI I felt challenged to do my best work.	3.2	4.2
2. I was concerned that I might not be understanding the material.	3.8	3.2
3. I was not concerned when I missed a question because no one was watching me anyway.	2.6	4.2
4. While taking CAI I felt isolated and alone.	4.6	3.8
5. I felt uncertain as to my performance in the programming course relative to the performance of others.	4.2	3.0
6. I found myself just trying to get through the material rather than trying to learn.	4.0	3.6
7. I knew whether my answer was correct or not before I was told.	3.8	3.6
8. I guessed at the answers to questions.	4.2	3.4
9. In a situation where I am trying to learn something, it is important to me to know where I stand relative to others.	3.6	2.0
10. As a result of having studied some material by CAI, I am interested in trying to find out more about the subject matter.	4.2	3.8
11. I was more involved in running the machine than in understanding the material.	4.2	4.2
12. I felt I could work at my own pace with CAI.	4.2	3.6
13. CAI makes the learning too mechanical.	3.4	3.0

TABLE 14--Continued

14. I felt as if I had a private tutor while on CAI.	3.0	3.0
15. I was aware of efforts to suit the material specifically to me.	1.8	2.2
16.* I found it difficult to concentrate on the course material because of the hardware.	4.4	3.2
17. Questions were asked which I felt were not relevant to the material presented	4.0	4.4
18. CAI is an inefficient use of the student's time.	4.4	3.4
19. While on CAI I encountered mechanical malfunctions.	2.4	2.2
20.* CAI made it possible for me to learn quickly.	4.2	2.6
21. I felt frustrated by the CAI situation.	3.8	3.4
22.* The CAI approach is inflexible.	4.4	3.2
23. Even otherwise interesting material would be boring when presented by CAI	4.4	3.6
24. In view of the effort I put into it, I was satisfied with what I learned while taking CAI.	4.4	3.6
25. In view of the amount I learned, I would say CAI is superior to traditional instruction.	3.2	2.2
26. With a course such as I took by CAI, I would prefer CAI to traditional instruction.	3.4	2.6

* significant difference at the .05 probability level

TABLE 14--Continued

27. I am not in favor of CAI because it is just another step toward depersonalized instruction.	4.4	4.2
28. CAI is too fast.	4.4	4.0
29. Typing experience is necessary in order to perform easily on CAI.	3.2	3.8
30. CAI is boring.	4.0	3.6

Only five of the possible 60 responses from the two CAI subgroups were negative toward CAI. The CAI-teletype subgroup scored negatively toward CAI on question 9, importance of knowing their standing relative to others in a class and question 25, superiority of CAI to traditional methods. The CAI-batch subgroup scored negatively toward CAI on question 19, encountered equipment malfunctions. The linear nature of the CAI course was manifest as both subgroups scored below 2.5 on question 15, "I was aware of efforts to suit the material specifically to me." As with the Semantic Differential ratings, analysis of responses in Table 14 could go on almost indefinitely; however, further analysis will not be performed in this paper.

To facilitate comparison of the results from the Brown scale obtained in this study with the results obtained by Mathis, Smith and Hansen [21], Table 15 presents the results of the Brown scale considering the CAI group as a single group rather than as two subgroups and lists the results from the other study mentioned above for comparison.

In most instances, the results of this study replicate the findings of Mathis, Smith and Hansen, as may be determined from examination of the table above. The correlation across 30 items between the means in the two studies is .64.

TABLE 15

COMPARISON OF BROWN SCALE RESULTS

Item	Means This Study	Mathis Study	S.D. This Study	Mathis Study
1	3.6	3.63	1.34	.07
2	3.5	2.75	.10	.16
3	3.3	3.88	1.50	.94
4	4.2	4.06	.78	.37
5	3.6	3.94	1.34	.11
6	3.8	3.94	.78	.91
7	3.7	3.59	.67	1.01
8	3.8	3.72	.78	1.11
9	3.0	3.22	1.24	1.12
10	4.0	3.63	.47	1.01
11	4.2	4.09	.63	.82
12	3.9	3.84	1.10	1.08
13	3.2	3.69	1.03	1.00
14	3.0	3.72	.94	1.17
15	2.0	3.16	.66	1.02
16	3.8	4.00	.91	.88
17	4.2	4.50	.91	.62
18	3.9	3.69	1.10	1.09
19	2.6	3.42	.84	1.54
20	3.4	3.41	1.17	.91
21	3.6	3.53	1.07	1.14
22	3.8	3.32	.91	1.04
23	4.0	4.09	.81	.82
24	4.0	3.66	.94	.87
25	2.7	3.03	.94	1.06
26	3.0	3.09	1.15	1.09
27	4.3	3.81	.48	.90
28	4.2	3.81	.42	.82
29	3.5	3.66	1.08	.94
30	3.8	3.97	1.03	.69

The teletype questionnaire was given to the teletype subgroups of both main groups. (See Appendix H.) There was no significant difference between the subgroups on any of the fourteen questions; however, a summary of the average responses made by the entire teletype group, that is, the two teletype subgroups combined, revealed some valuable information.

The students in the teletype group felt it was a little more beneficial to execute their program assignments on the teletype as opposed to batch mode. They finished their program assignments slightly before the due date. The teletype group had a little trouble scheduling time to use a teletype, and they felt that the teletype moderately increased their rate of progress. They also felt that the noise of the teletype had a small negative effect on their concentration. Having other people in the room during a teletype session did not bother them and sometimes tended to help the students in the teletype group. The students in the teletype group lost a moderate number of files due to RESPOND system malfunctions. They had practically no trouble learning to use RESPOND. Most students used the teletype in the afternoon or early evening and the average duration of a session was 1-1/2 hours. It took 3 to 6 runs of a program assignment to get it debugged. The teletype group did not use paper tape or punch cards from the teletype to any great degree. The students in the teletype group felt that the amount of disk space available to them (20 blocks) was not nearly enough for manipulating files in the RESPOND time-sharing system.

The PICLS system that was used in this study had a number of shortcomings relative to its effectiveness, convenience and generality as an integral part of the instructional situation. The absence of record keeping facilities hampered the collection of student responses while the absence of a restart mechanism made the continuance of instruction from session to session less than optimum in some instances. It should be mentioned that record keeping facilities were included in the PICLS system obtained from Purdue but were not implemented in the version used at Texas due partly to the experimental nature of this CAI effort on the CDC 6600 and due partly to the lack of sufficient time for implementation.

Due to the environment in which PICLS executed and its dependence upon RESPOND and the CONVERSE feature, constraints such as the relatively short time limit allowed a conversational job and the relatively inefficient organization and manipulation of the files that constituted the CAI course made working with the system less convenient than might have otherwise been the case. Desirable features which should be considered for incorporation into PICLS include the ability to trap alphanumeric responses made by the student in order to perform more extensive answer processing; provisions for allowing the course author to work with a seemingly unrestricted course structure, lesson size and time limit; and provisions for allowing the student to input multiple-line answers such as a sequence of program code. In particular the ability to trap responses might provide answer processing which utilized an interpreter to execute and validate answers consisting of code segments.

Using the teletypewriter in this study provided an adequate but less than desirable instructional interface device. The slowness of the teletype and the inherent noise of the machine seem to be the most frustrating and irritating features from the student's point of view. This particular course seemed to amplify the slow type-out rate of the teletype due to the design of the course. The CAI course was intended to be a self-contained entity. In order to achieve this property of being self-contained, there are numerous instances throughout the course where a relatively long type-out is presented to the student before a response is requested. Much of the information contained in these long type-outs could perhaps be presented by means of the ELASTIC manual [17] and other supplemental material if it was determined that a self-contained course were no longer desired. The cost of the terminal time used on the CDC 6600 while a student read a manual or text rather than receiving the information via teletype is normally no more than the cost of the telephone line charge. The same cost situation does not exist for all CAI systems. Rewriting the course in such a way as to reduce long type-outs might prove to be a valuable revision.

The use of a CRT terminal instead of a teletype would remove the problem of noise; however, the hard copy that results from a session at the teletype would not normally be available. Some CRT terminals might also provide faster-than-teletype output rates. An additional constraint that exists when using a CRT terminal is the increased cost involved. For example, approximately 5-1/2 teletypes can be had for the same cost as one Data Point 3300 display terminal.

From the experiences of this study, a terminal with the following properties would be most desirable for teaching programming languages: graphic display screen with optional hard copy output, typewriter keyboard for input, rapid output rate at least three times faster than a teletype, noiseless, facility for switching between instruction and execution of programs in a language by depressing a single switch or button, and finally a light pen for use in editing responses and programs as well as interaction with the instructional system. In fact, if it weren't for the high cost, a CDC 250 Display Console would make an ideal student terminal.

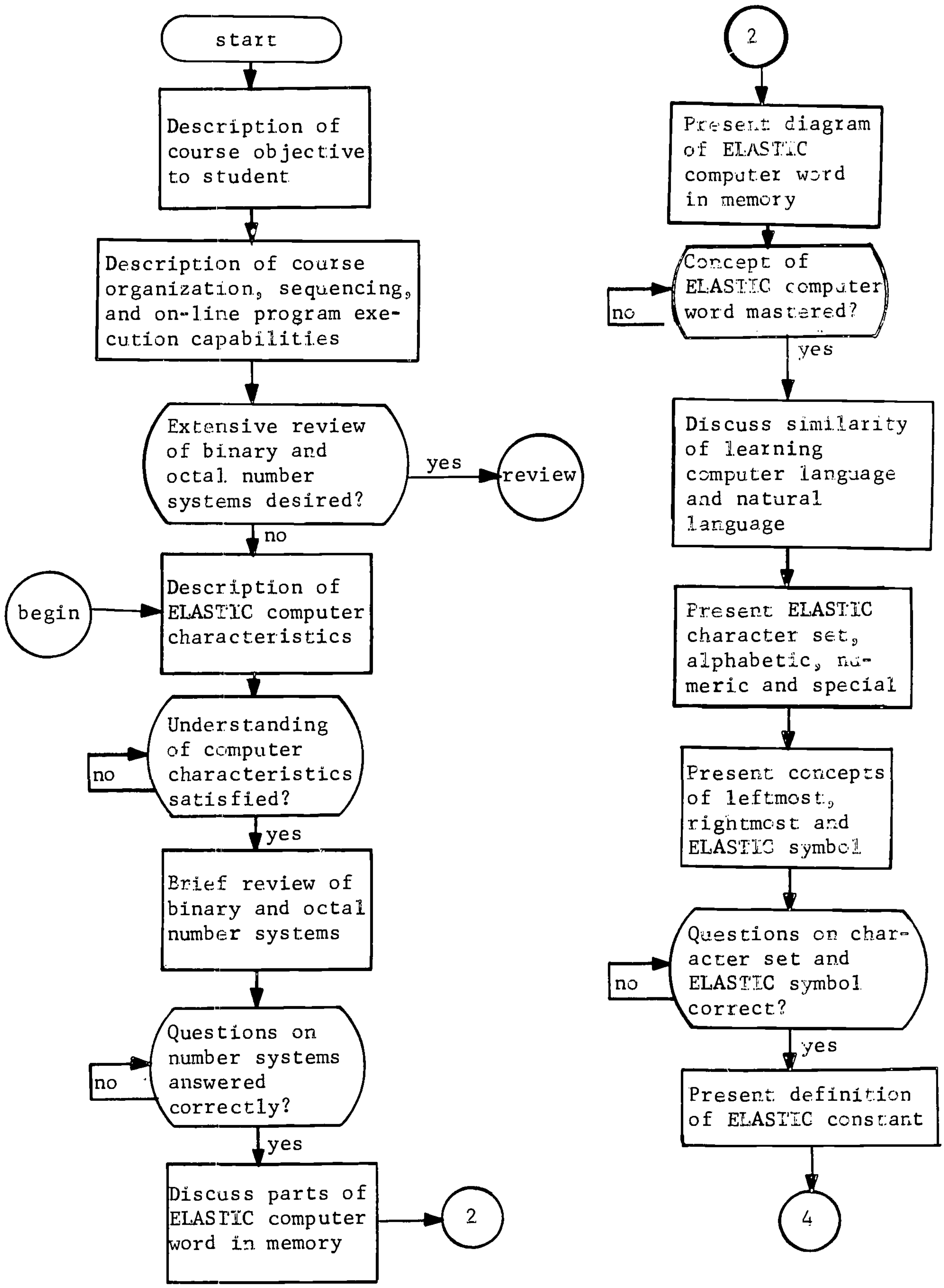
In conclusion, it would appear that one of the more important findings of this experiment was that providing a student with an instructional environment in which he is self-paced and has unrestricted use of the computer in interactive mode may result in a student who has more motivation for using the computer for problem solving and a greater degree of independence to work alone without outside intervention. The behavioral evidence that students in the CAI-teletype group ran a surprisingly greater number of jobs through the computer and spent many more hours at the teletype terminal than students in any of the other three groups indicates that desirable side effects of the experience can be achieved. Students in the teletype groups also rated the computer as being more interesting, active, flexible and strong than students in the batch groups. Finally, the fact that all students in the CAI groups want to go on to further study in computer science is impressive.

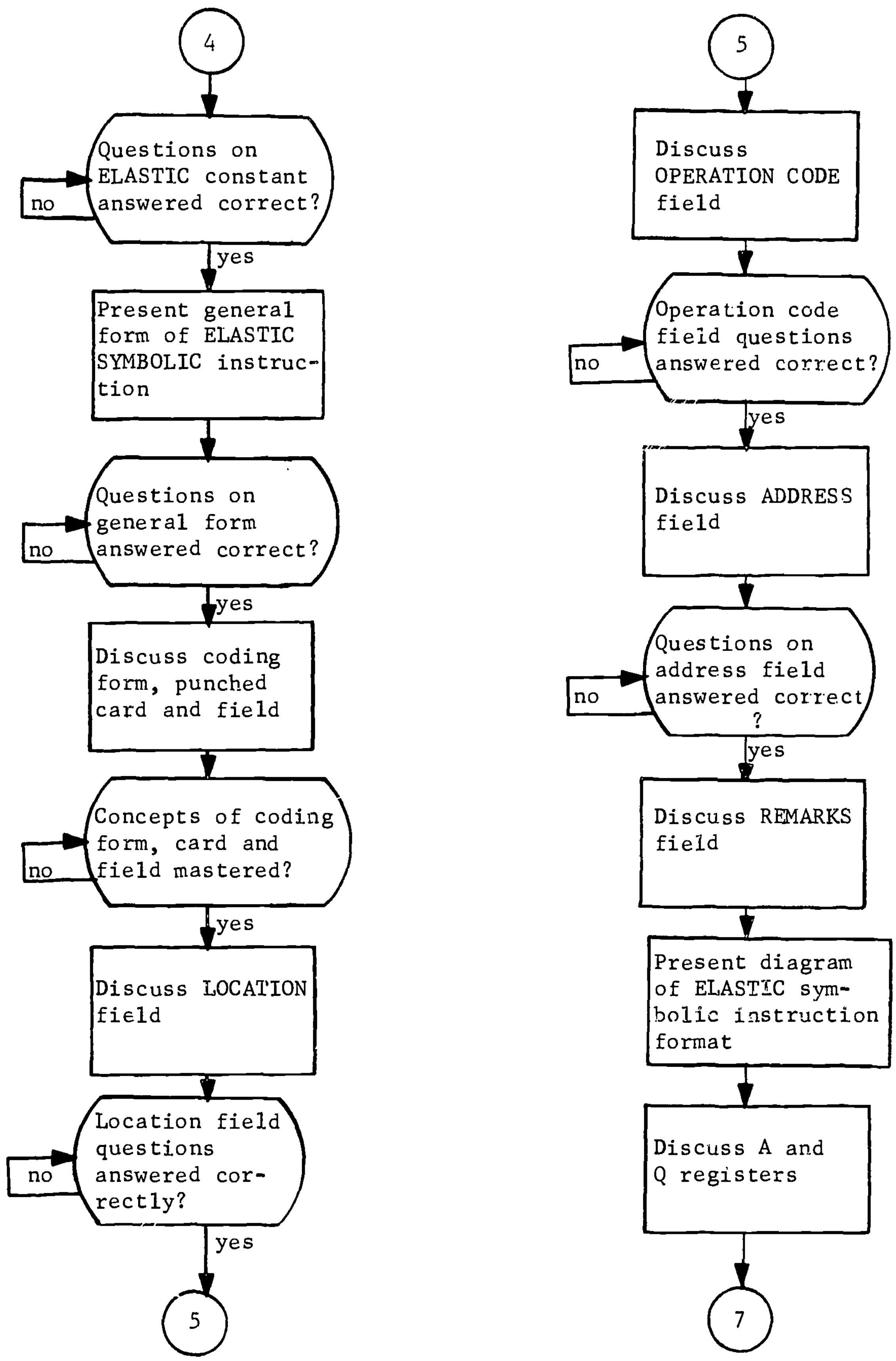
APPENDIX A

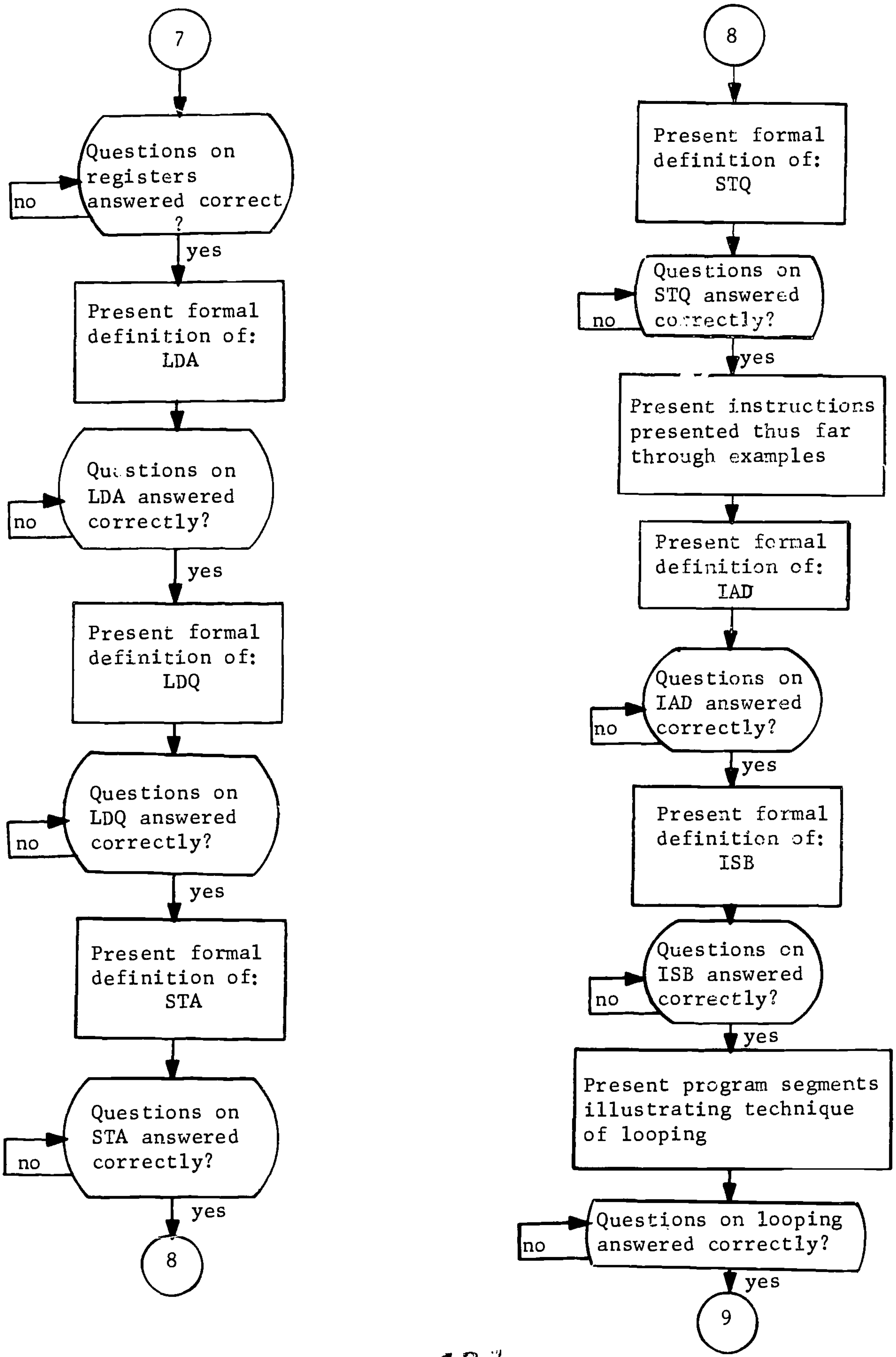
FLOWCHART AND STRUCTURE OF INSTRUCTIONAL PROGRAM

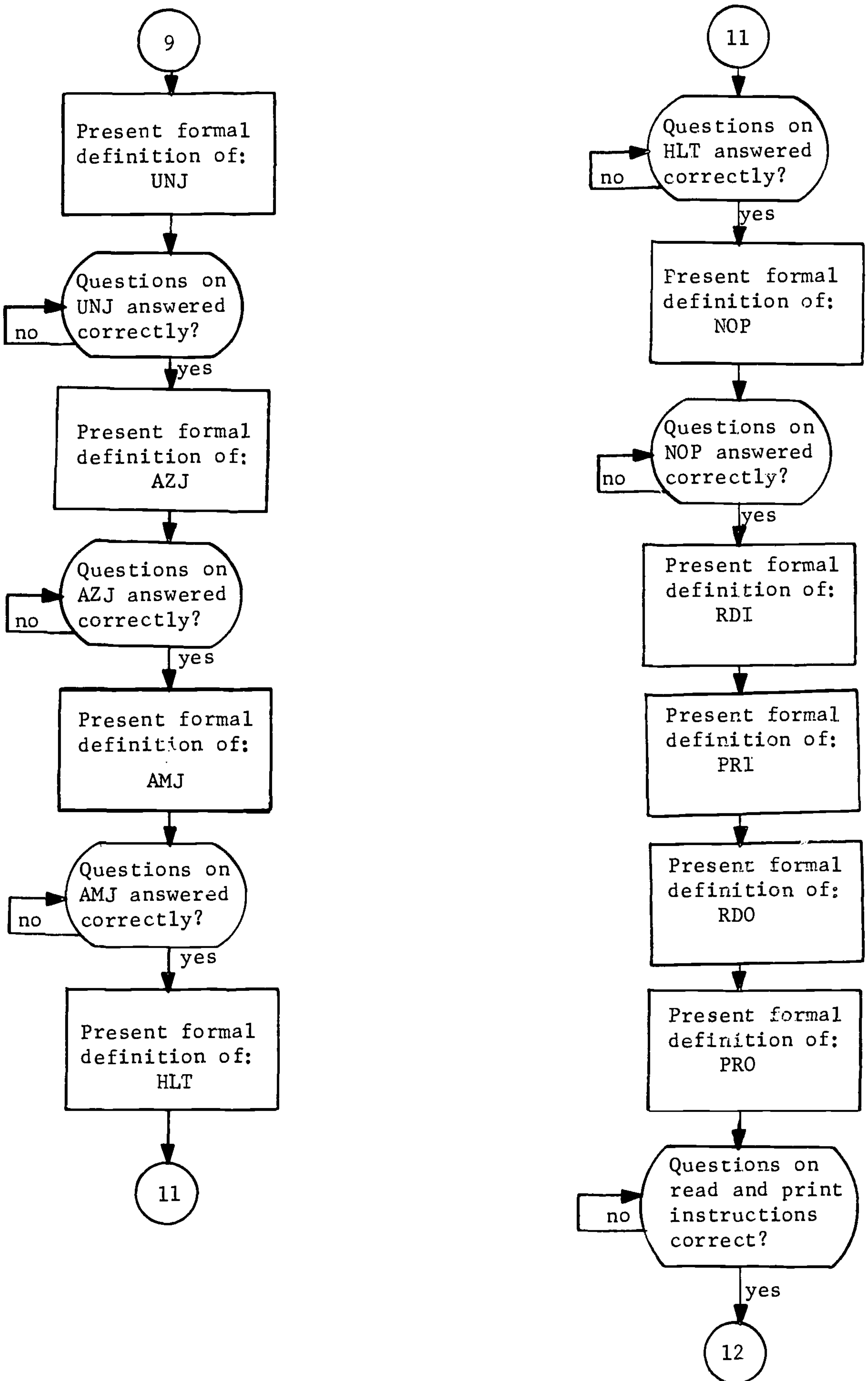
A flowchart of the instructional program is presented in the following pages. The course structure is linear in nature and thus all students take very similar paths through the course material. The fact that the course is divided into eighteen subcourses is not reflected in this flowchart due to the fact that the flowchart is intended to be machine independent while the course subdividing procedure was dictated by implementation on a particular instructional system.

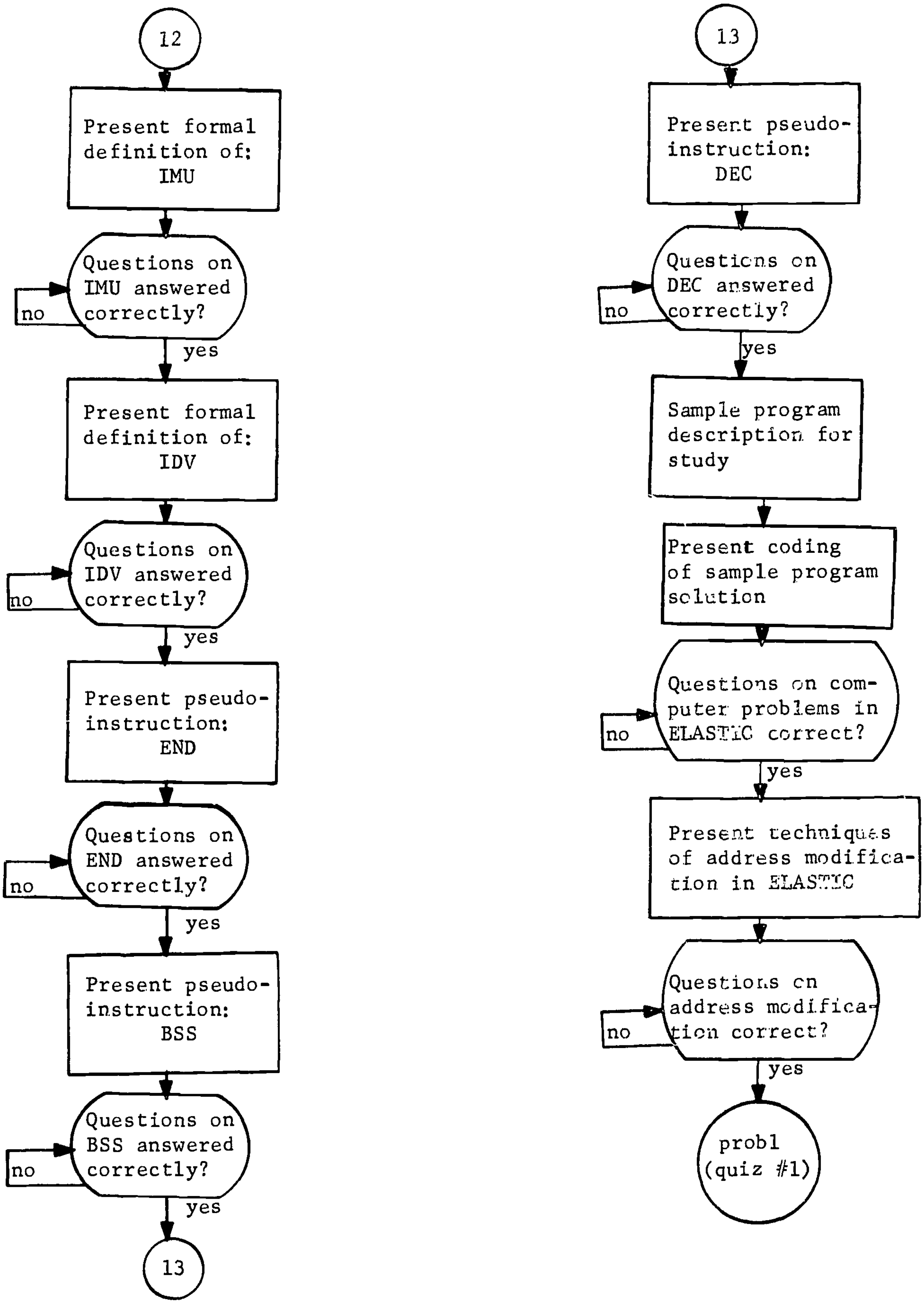
The special flowchart symbol "$\boxed{\text{no}}$" is used to indicate the fact that the student is normally given two chances to give the correct answer to a question or problem. After two chances the correct answer is presented and the student is asked to type it in.

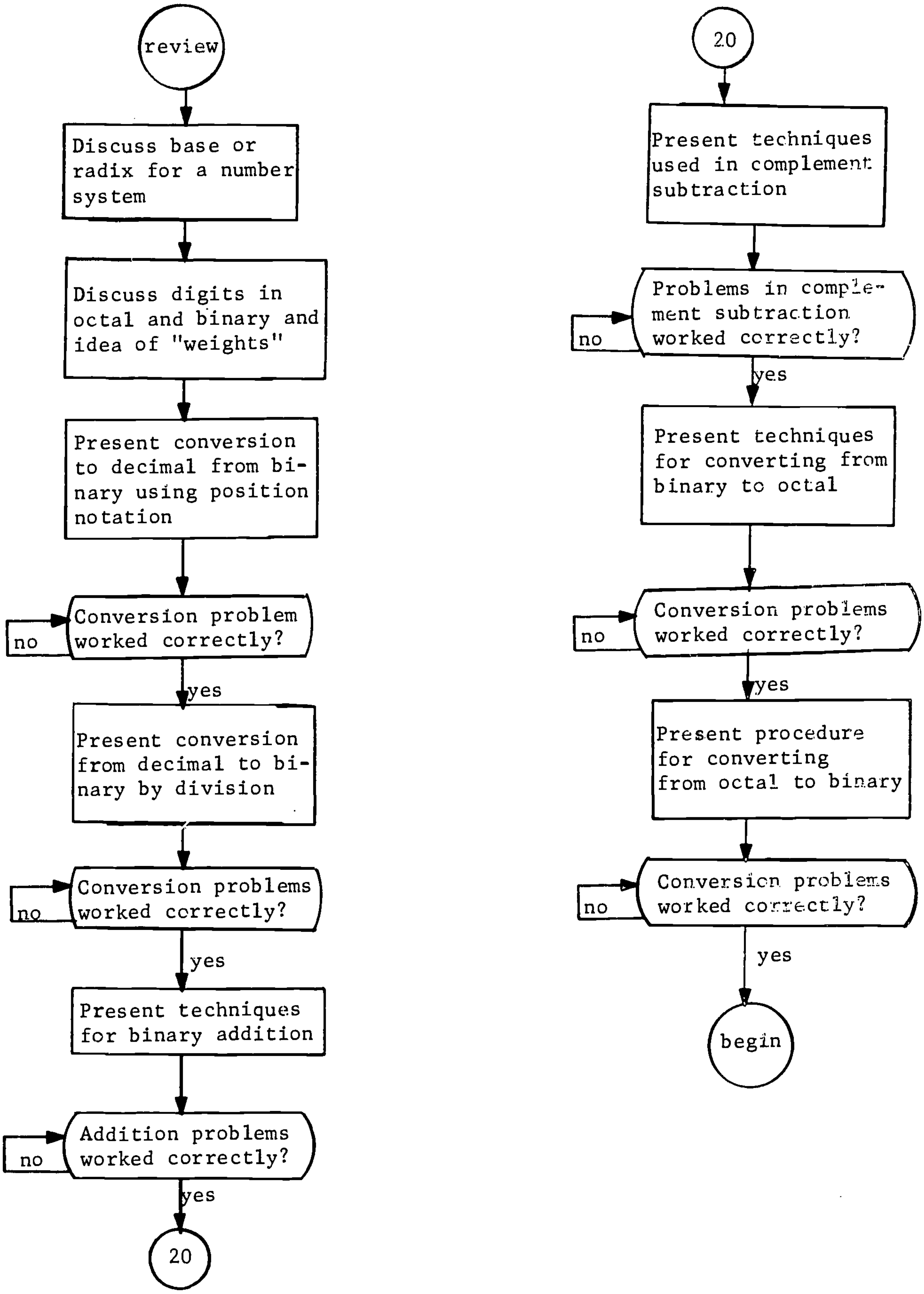

start
Description of course objective to student
Description of course organization, sequencing, and on-line program execution capabilities
Extensive review of binary and octal number systems desired?
yes
review
no
begin
Description of ELASTIC computer characteristics
Understanding of computer characteristics satisfied?
no
yes
Brief review of binary and octal number systems
Questions on number systems answered correctly?
no
yes
Discuss parts of ELASTIC computer word in memory
2
2
Present diagram of ELASTIC computer word in memory
Concept of ELASTIC computer word mastered?
no
yes
Discuss similarity of learning computer language and natural language
Present ELASTIC character set, alphabetic, numeric and special
Present concepts of leftmost, rightmost and ELASTIC symbol
Questions on character set and ELASTIC symbol correct?
no
yes
Present definition of ELASTIC constant
4


4
Questions on ELASTIC constant answered correct?
no
yes
Present general form of ELASTIC SYMBOLIC instruction
Questions on general form answered correct?
no
yes
Discuss coding form, punched card and field
Concepts of coding form, card and field mastered?
no
yes
Discuss LOCATION field
Location field questions answered correctly?
no
yes
5
5
Discuss OPERATION CODE field
Operation code field questions answered correct?
no
yes
Discuss ADDRESS field
Questions on address field answered correct ?
no
yes
Discuss REMARKS field
Present diagram of ELASTIC symbolic instruction format
Discuss A and Q registers
7

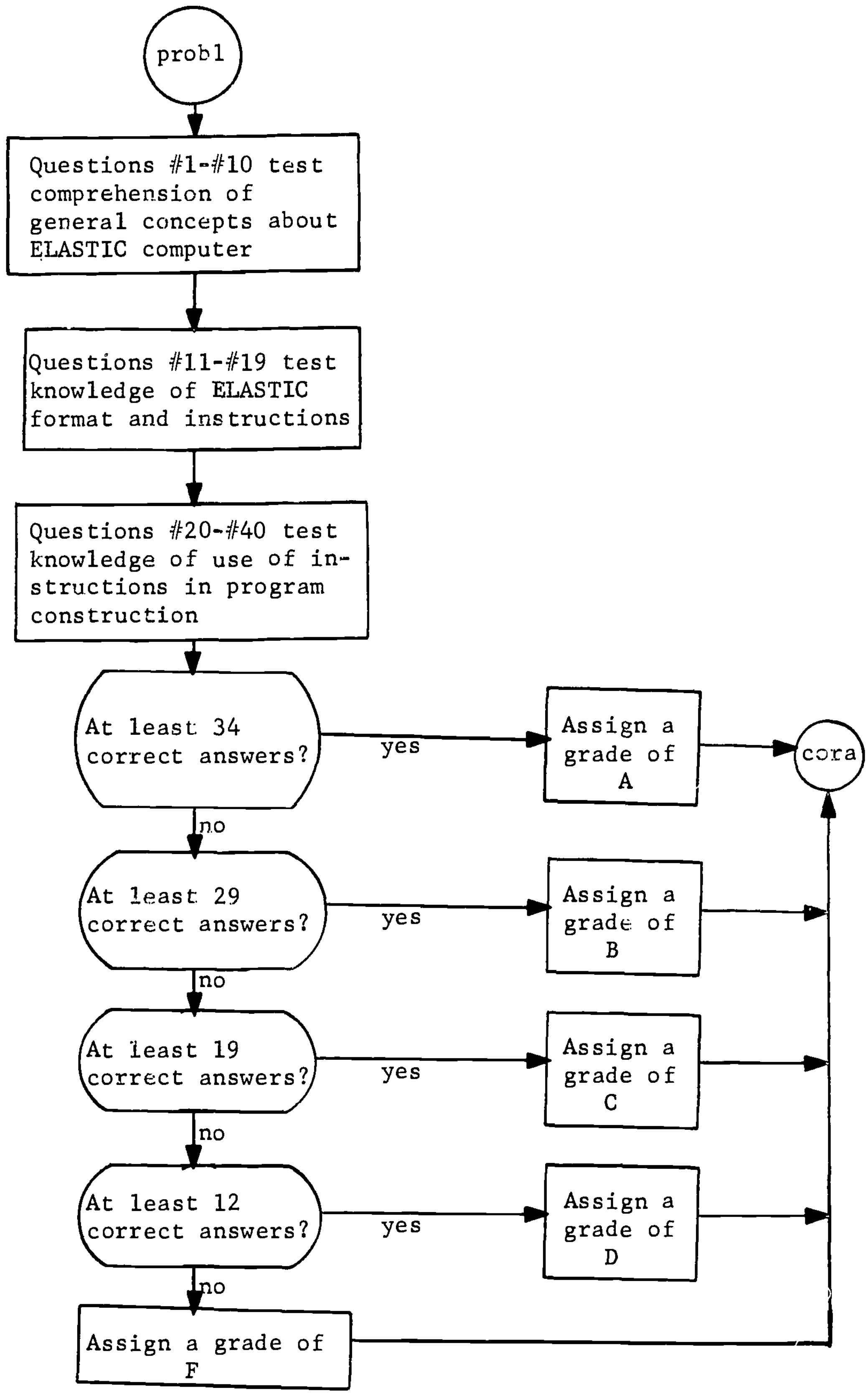

7
Questions on registers answered correct ?
no
yes
Present formal definition of: LDA
Questions on LDA answered correctly?
no
yes
Present formal definition of: LDQ
Questions on LDQ answered correctly?
no
yes
Present formal definition of: STA
Questions on STA answered correctly?
no
yes
8
8
Present formal definition of: STQ
Questions on STQ answered correctly?
no
yes
Present instructions presented thus far through examples
Present formal definition of: IAD
Questions on IAD answered correctly?
no
yes
Present formal definition of: ISB
Questions on ISB answered correctly?
no
yes
Present program segments illustrating technique of looping
Questions on looping answered correctly?
no
yes
9

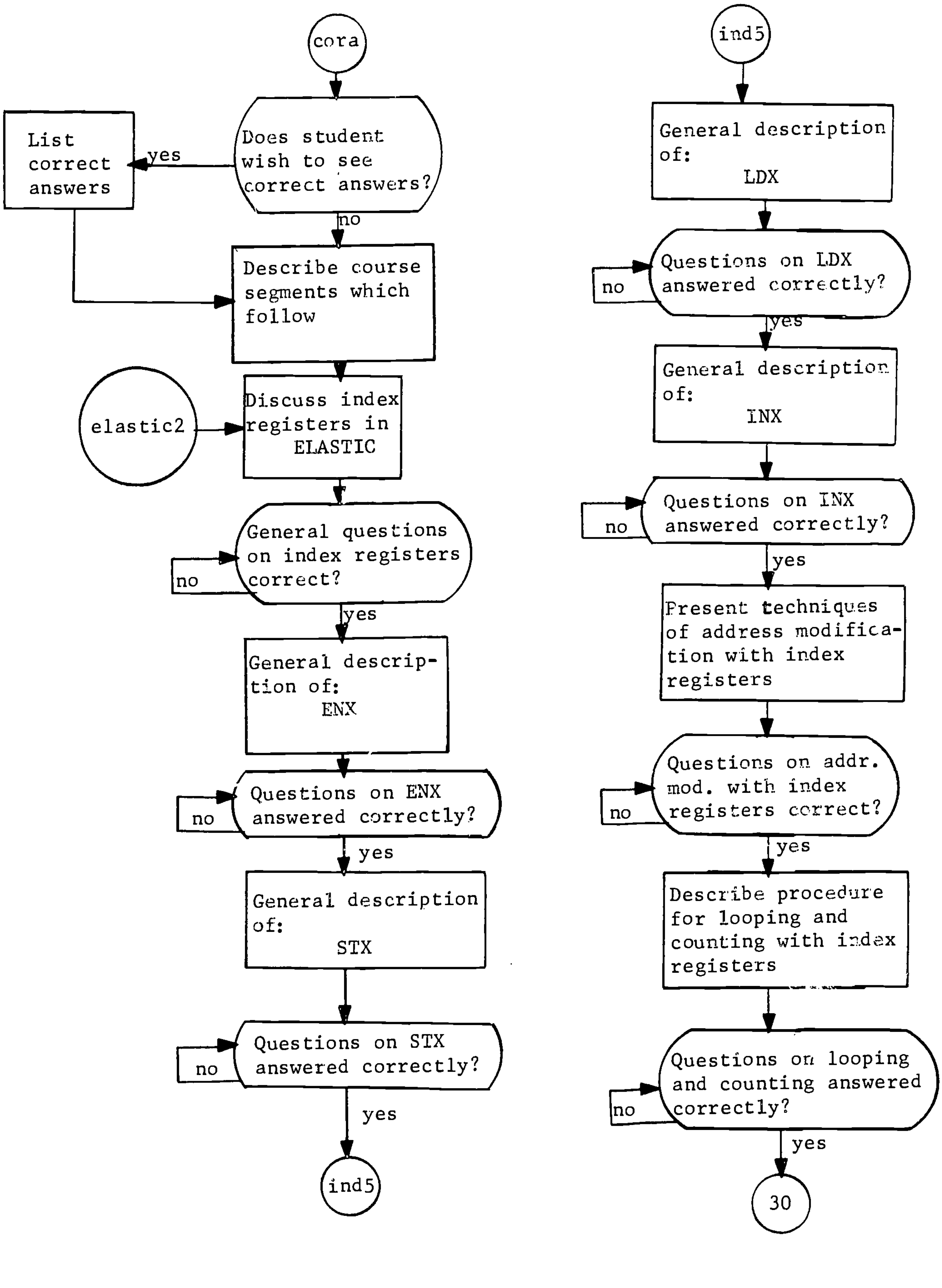

9
Present formal definition of: UNJ
Questions on UNJ answered correctly?
no
yes
Present formal definition of: AZJ
Questions on AZJ answered correctly?
no
yes
Present formal definition of: AMJ
Questions on AMJ answered correctly?
no
yes
Present formal definition of: HLT
11
11
Questions on HLT answered correctly?
no
yes
Present formal definition of: NOP
Questions on NOP answered correctly?
no
yes
Present formal definition of: RDI
Present formal definition of: PRI
Present formal definition of: RDO
Present formal definition of: PRO
Questions on read and print instructions correct?
no
yes
12

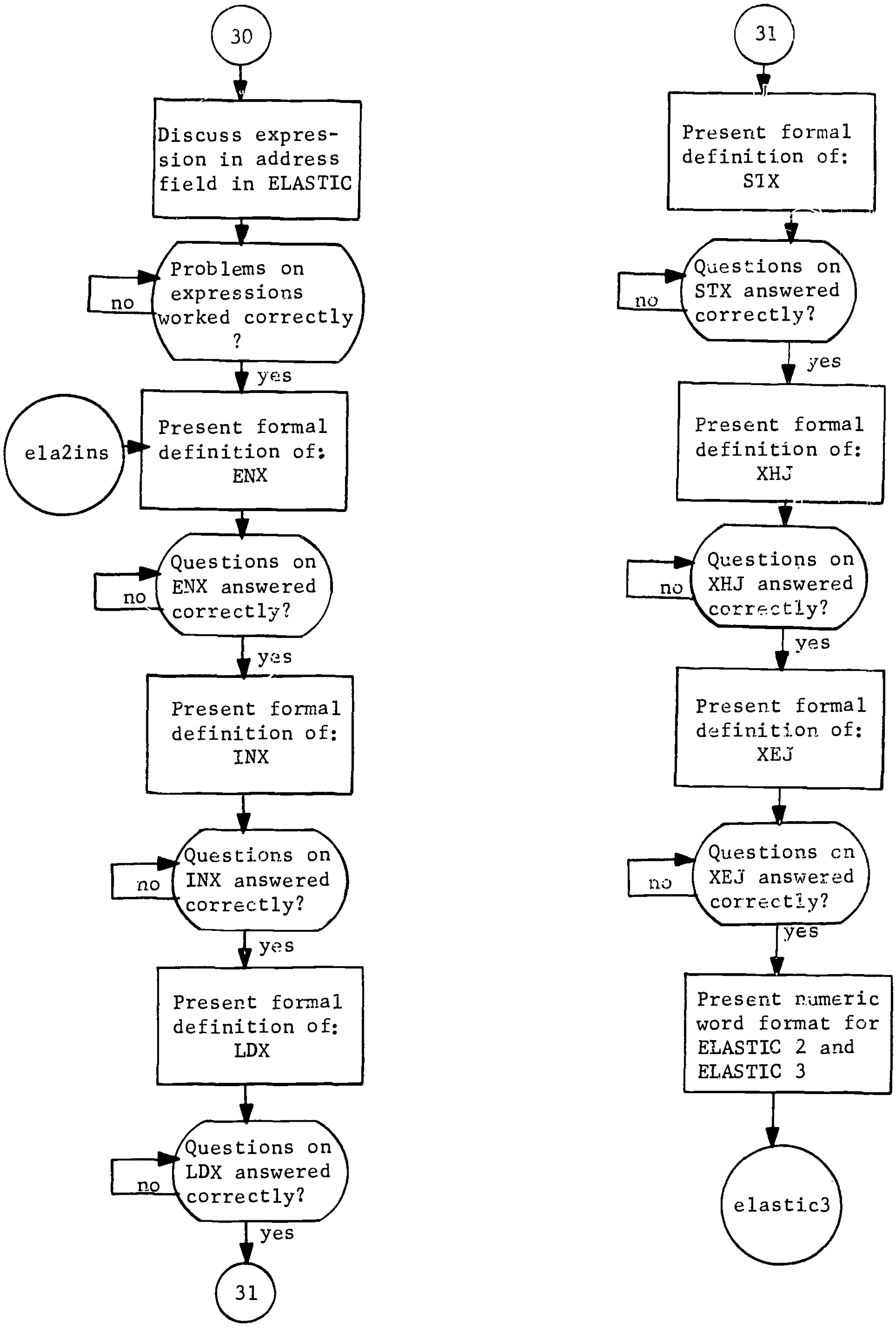

12
Present formal definition of: IMU
Questions on IMU answered correctly?
no
yes
Present formal definition of: IDV
Questions on IDV answered correctly?
no
yes
Present pseudo-instruction: END
Questions on END answered correctly?
no
yes
Present pseudo-instruction: BSS
Questions on BSS answered correctly?
no
yes
13
13
Present pseudo-instruction: DEC
Questions on DEC answered correctly?
no
yes
Sample program description for study
Present coding of sample program solution
Questions on computer problems in ELASTIC correct?
no
yes
Present techniques of address modification in ELASTIC
Questions on address modification correct?
no
yes
probl (quiz #1)

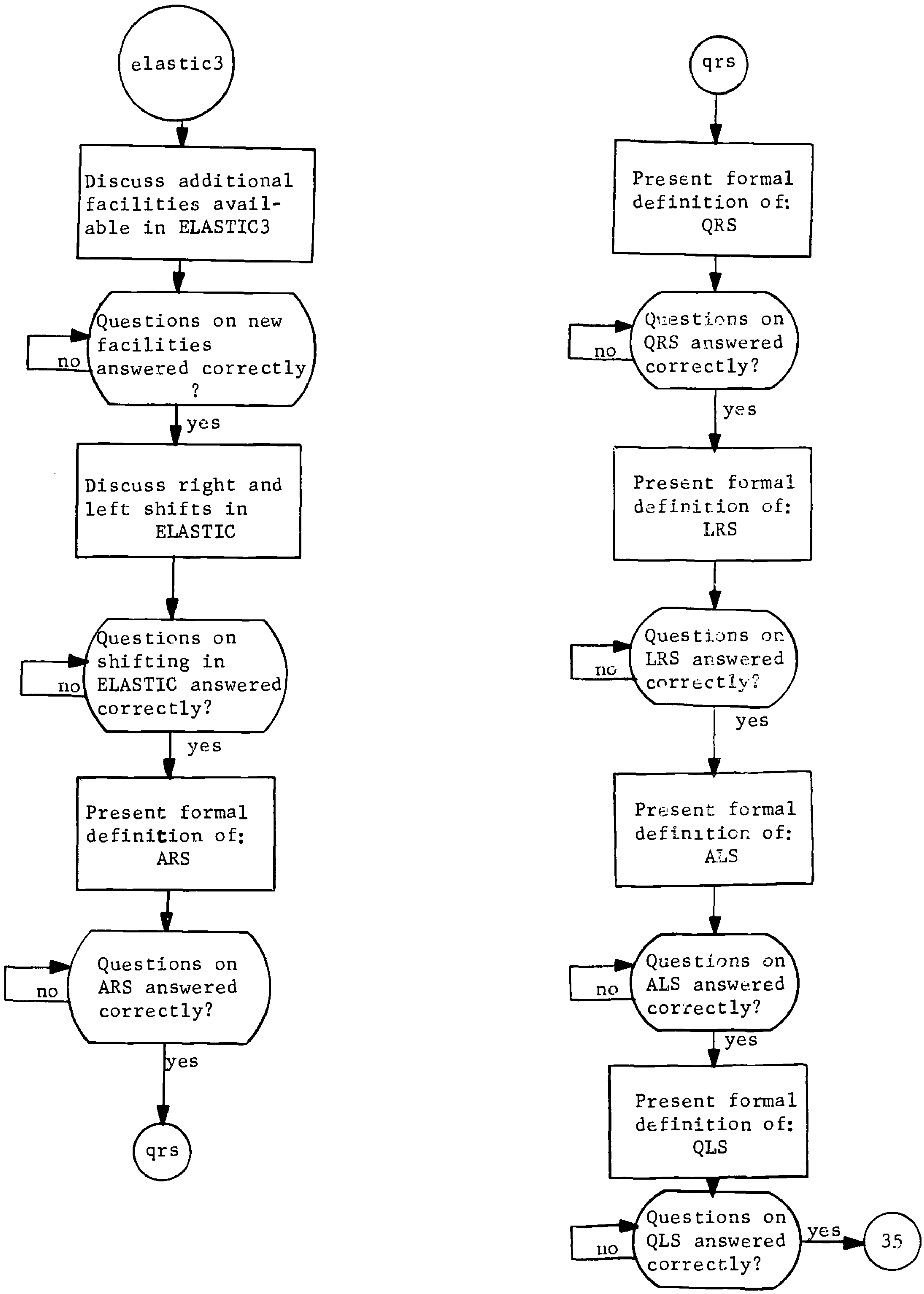

review
Discuss base or radix for a number system
Discuss digits in octal and binary and idea of "weights"
Present conversion to decimal from binary using position notation
Conversion problem worked correctly?
no
yes
Present conversion from decimal to binary by division
Conversion problems worked correctly?
no
yes
Present techniques for binary addition
Addition problems worked correctly?
no
yes
20
20
Present techniques used in complement subtraction
Problems in complement subtraction worked correctly?
no
yes
Present techniques for converting from binary to octal
Conversion problems worked correctly?
no
yes
Present procedure for converting from octal to binary
Conversion problems worked correctly?
no
yes
begin

Probl constitutes a quiz consisting of 40 questions which test the student's comprehension of the ideas presented in ELASTIC 1 and general computer concepts.

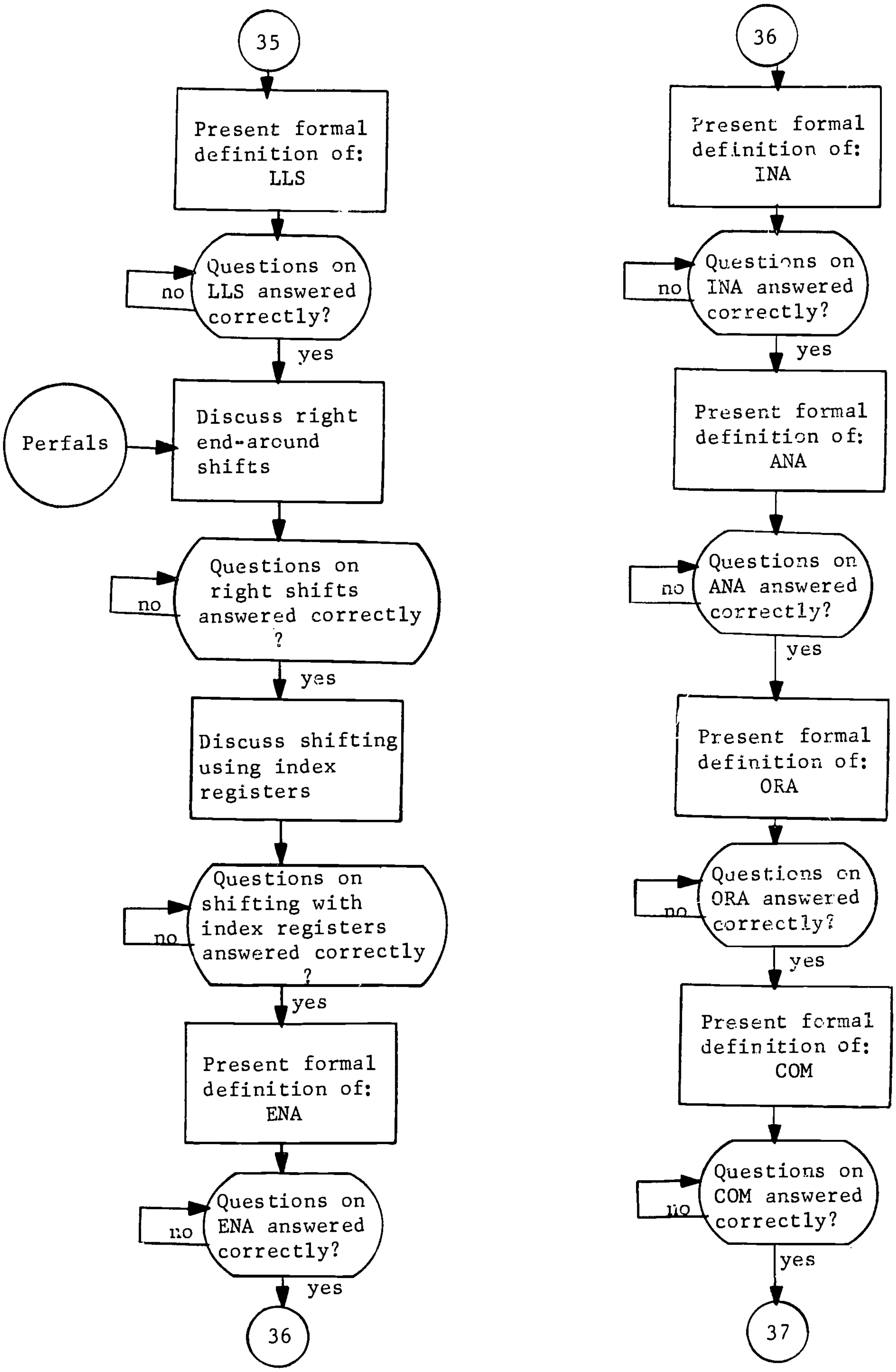


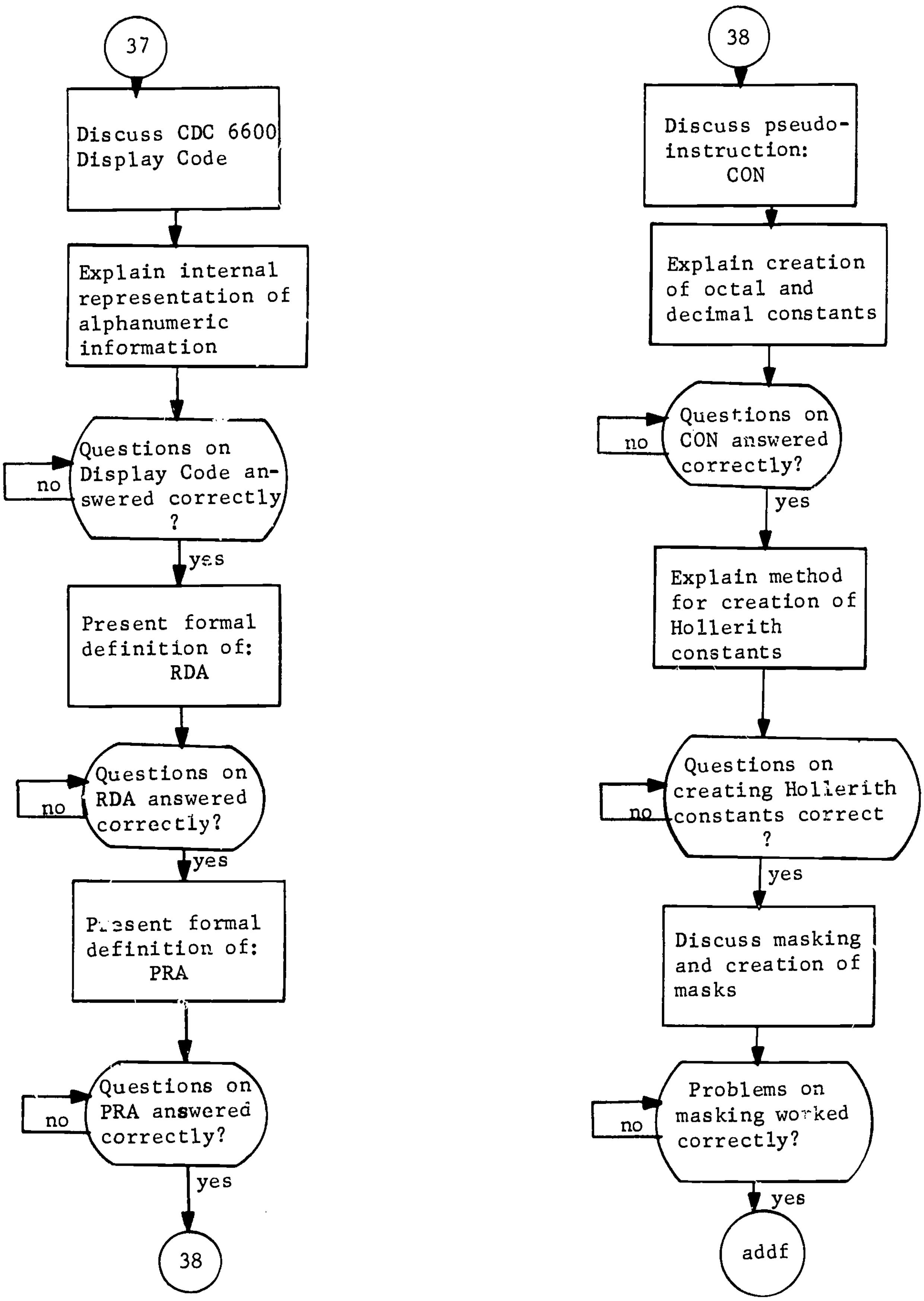

cora
Does student wish to see correct answers?
yes
List correct answers
no
Describe course segments which follow
elastic2
Discuss index registers in ELASTIC
General questions on index registers correct?
no
yes
General description of: ENX
Questions on ENX answered correctly?
no
yes
General description of: STX
Questions on STX answered correctly?
no
yes
ind5
ind5
General description of: LDX
Questions on LDX answered correctly?
no
yes
General description of: INX
Questions on INX answered correctly?
no
yes
Present techniques of address modification with index registers
Questions on addr. mod. with index registers correct?
no
yes
Describe procedure for looping and counting with index registers
Questions on looping and counting answered correctly?
no
yes
30

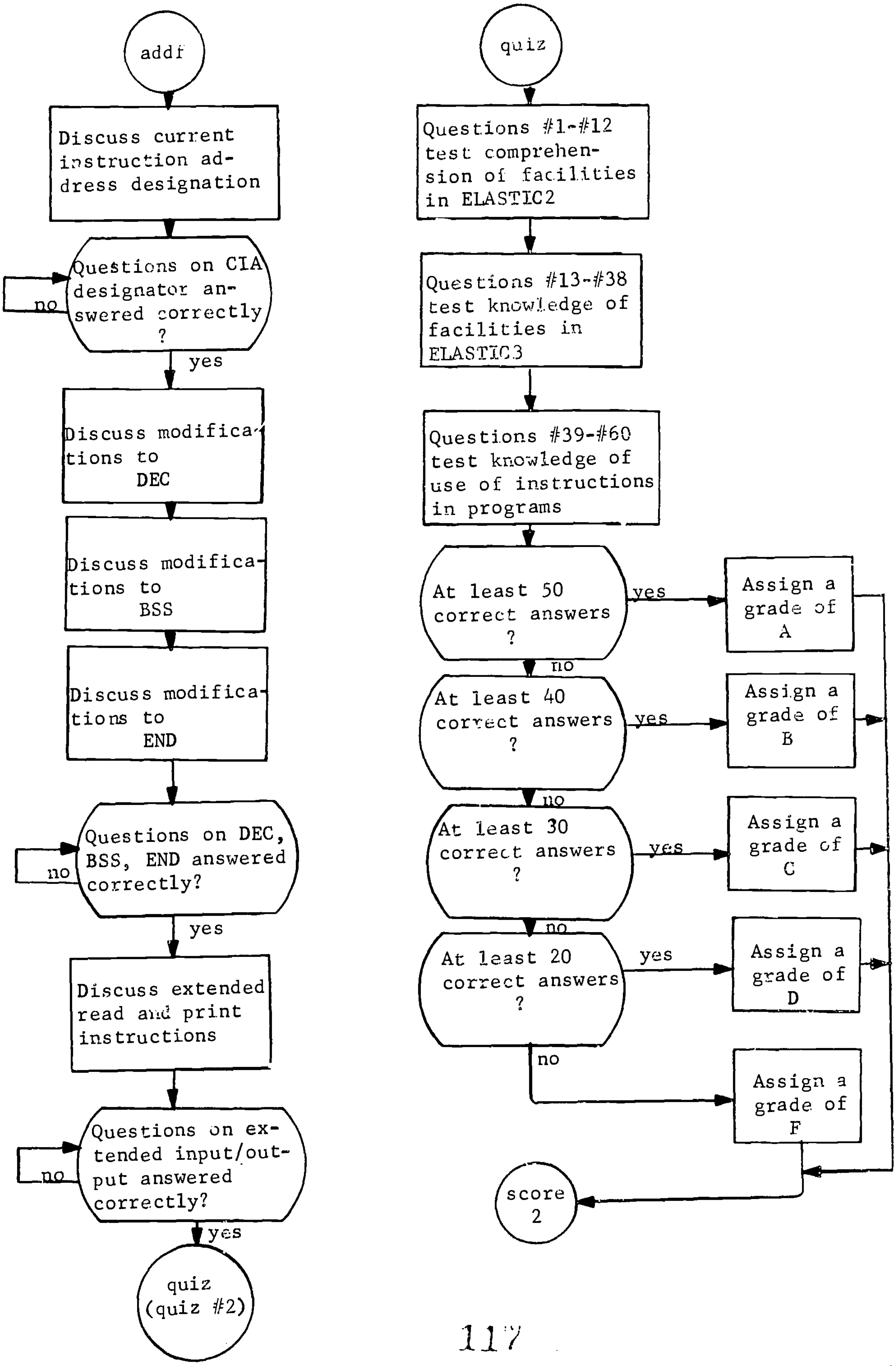

30
Discuss expression in address field in ELASTIC
Problems on expressions worked correctly ?
no
yes
ela2ins
Present formal definition of: ENX
Questions on ENX answered correctly?
no
yes
Present formal definition of: INX
Questions on INX answered correctly?
no
yes
Present formal definition of: LDX
Questions on LDX answered correctly?
no
yes
31
31
Present formal definition of: STX
Questions on STX answered correctly?
no
yes
Present formal definition of: XHJ
Questions on XHJ answered correctly?
no
yes
Present formal definition of: XEJ
Questions on XEJ answered correctly?
no
yes
Present numeric word format for ELASTIC 2 and ELASTIC 3
elastic3



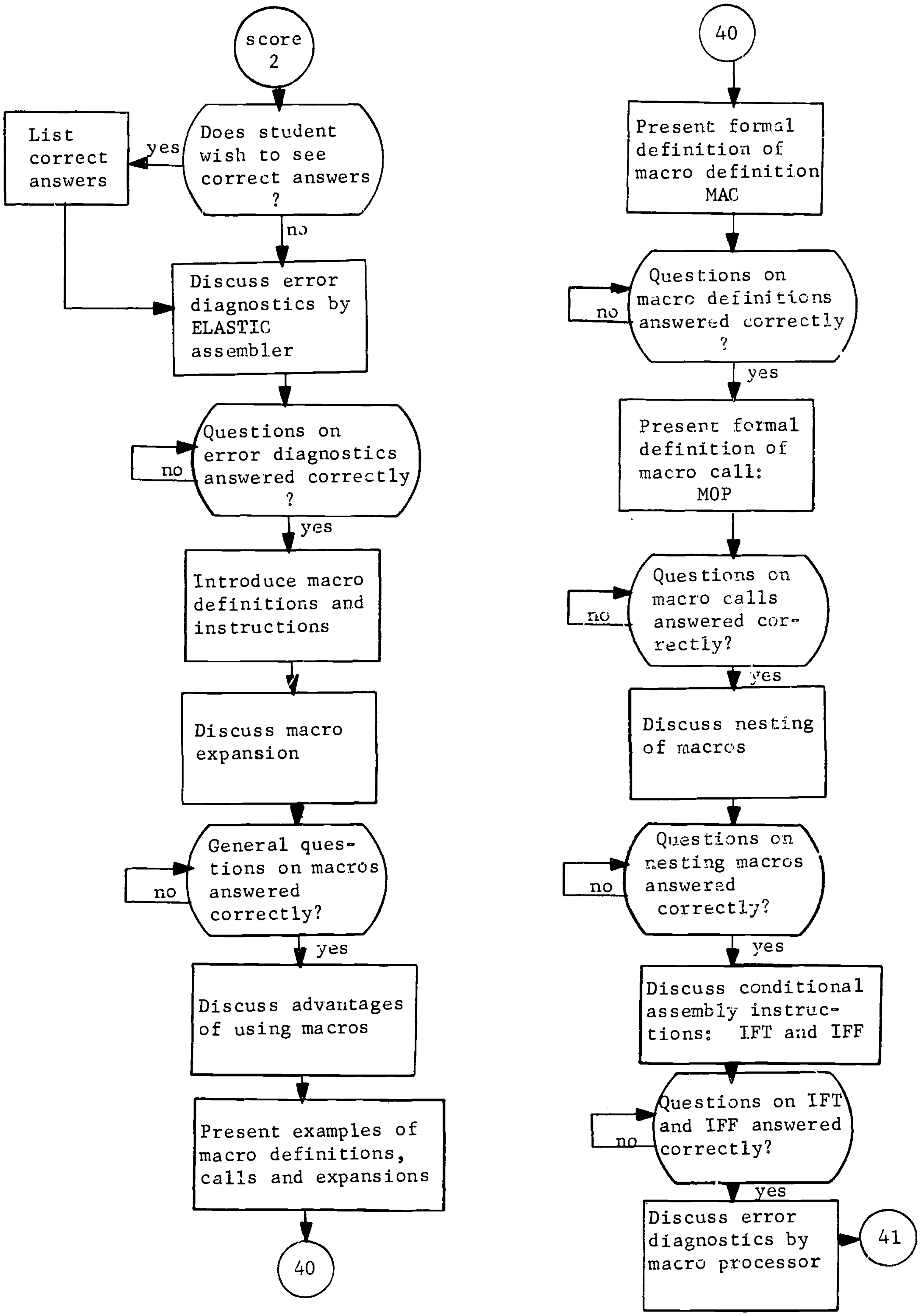
elastic3
Discuss additional facilities available in ELASTIC3
Questions on new facilities answered correctly ?
no
yes
Discuss right and left shifts in ELASTIC
Questions on shifting in ELASTIC answered correctly?
no
yes
Present formal definition of: ARS
Questions on ARS answered correctly?
no
yes
qrs
qrs
Present formal definition of: QRS
Questions on QRS answered correctly?
no
yes
Present formal definition of: LRS
Questions on LRS answered correctly?
no
yes
Present formal definition of: ALS
Questions on ALS answered correctly?
no
yes
Present formal definition of: QLS
Questions on QLS answered correctly?
no
yes
35

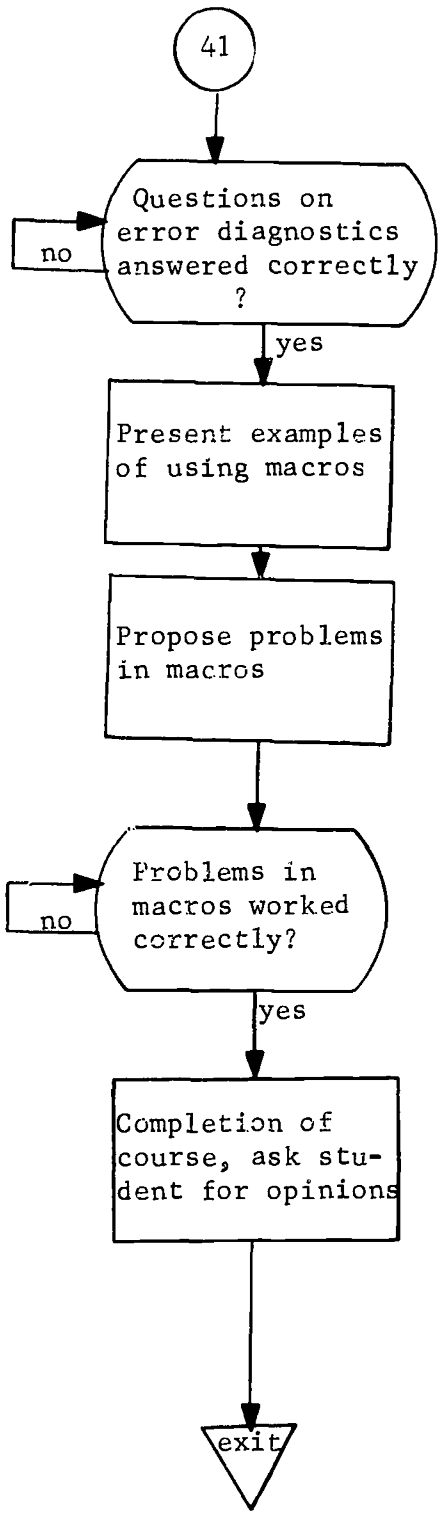

35
Present formal definition of: LLS
Questions on LLS answered correctly?
no
yes
Perfals
Discuss right end-around shifts
Questions on right shifts answered correctly ?
no
yes
Discuss shifting using index registers
Questions on shifting with index registers answered correctly ?
no
yes
Present formal definition of: ENA
Questions on ENA answered correctly?
no
yes
36
36
Present formal definition of: INA
Questions on INA answered correctly?
no
yes
Present formal definition of: ANA
Questions on ANA answered correctly?
no
yes
Present formal definition of: ORA
Questions on ORA answered correctly?
no
yes
Present formal definition of: COM
Questions on COM answered correctly?
no
yes
37


37
Discuss CDC 6600 Display Code
Explain internal representation of alphanumeric information
Questions on Display Code answered correctly ?
no
yes
Present formal definition of: RDA
Questions on RDA answered correctly?
no
yes
Present formal definition of: PRA
Questions on PRA answered correctly?
no
yes
38
38
Discuss pseudo-instruction: CON
Explain creation of octal and decimal constants
Questions on CON answered correctly?
no
yes
Explain method for creation of Hollerith constants
Questions on creating Hollerith constants correct ?
no
yes
Discuss masking and creation of masks
Problems on masking worked correctly?
no
yes
addf



addf
Discuss current instruction address designation
Questions on CIA designator answered correctly ?
no
yes
Discuss modifications to DEC
Discuss modifications to BSS
Discuss modifications to END
Questions on DEC, BSS, END answered correctly?
no
yes
Discuss extended read and print instructions
Questions on extended input/output answered correctly?
no
yes
quiz (quiz #2)
quiz
Questions #1-#12 test comprehension of facilities in ELASTIC2
Questions #13-#38 test knowledge of facilities in ELASTIC3
Questions #39-#60 test knowledge of use of instructions in programs
At least 50 correct answers ?
yes
Assign a grade of A
no
At least 40 correct answers ?
yes
Assign a grade of B
no
At least 30 correct answers ?
yes
Assign a grade of C
no
At least 20 correct answers ?
yes
Assign a grade of D
no
Assign a grade of F
score 2


score 2
Does student wish to see correct answers ?
yes
List correct answers
no
Discuss error diagnostics by ELASTIC assembler
Questions on error diagnostics answered correctly ?
no
yes
Introduce macro definitions and instructions
Discuss macro expansion
General questions on macros answered correctly?
no
yes
Discuss advantages of using macros
Present examples of macro definitions, calls and expansions
40
40
Present formal definition of macro definition MAC
Questions on macro definitions answered correctly ?
no
yes
Present formal definition of macro call: MOP
Questions on macro calls answered correctly?
no
yes
Discuss nesting of macros
Questions on nesting macros answered correctly?
no
yes
Discuss conditional assembly instructions: IFT and IFF
Questions on IFT and IFF answered correctly?
no
yes
Discuss error diagnostics by macro processor
41


41
Questions on error diagnostics answered correctly ?
no
yes
Present examples of using macros
Propose problems in macros
Problems in macros worked correctly?
no
yes
Completion of course, ask student for opinions
exit

The instructional program was implemented using the PICLS language and instructional system. Using this language made it possible to execute the CAI program using a CDC 6600 computer system. The file management routines in PICLS placed constraints upon the physical organization of the instructional program which resulted in the division of the CAI course into eighteen subcourses. Each subcourse was further subdivided into sections or lessons.

The listing that follows constitutes a brief sketch of the instructional program as it exists for execution on the PICLS instructional system. Included in the listing of subcourses and sections is a short explanation of the instructional content or curriculum of the individual sections. The listing should provide a broad overview to the organization of the CAI course.

The form used in the listing is as follows:

"Subcourse"

    "Section"

        Description of contents of section named above

    "Section"

        Description of contents of section named above

"Subcourse"

  .
  .
  .
  etc.

ELAST01

    INTRO

        INTRODUCTION, HOW TO ENTER RESPONSES, ETC.

STRUCT

LIST OF SUBCOURSES AND SECTIONS

PRELIM

DESCRIPTION OF PREREQUISITES

ELAST02

REVIEW

BINARY NUMBERS

REV1

BINARY ARITHMETIC

OCT

BINARY AND OCTAL CONVERSIONS

ELAST03

BEGIN

COMMENTS ON ELASTIC COMPUTER

SYMB

DEFINITIONS: SYMBOL, CONSTANT, ETC.

OPR

CONCEPTS: OPERAND, OP CODE, ETC.

ELAST04

ELAST1A

CONCEPTS: ASSEMBLY LANGUAGE, CARD FORMAT, ETC.

REGIS

EXPLANATION OF A AND Q REGISTERS

ELAST05

ELATI

LDA,STA,IAD,ISB AND CODE SEGMENTS

ELATIC

UNJ,AZJ,AMJ,HLT,NOP AND CODE SEGMENTS

ELAST06

IO

RDI,PRI,RDO,PRO AND CODE SEGMENTS

ELATIB

IMU,IDV AND CODE SEGMENTS

PSDO

END,BSS,DEC AND CODE SEGMENTS

ELAST07

PROBI

SAMPLE ELASTIC PROGRAM

SAMP

PROGRAM SEGMENTS, LOOPING,ADDRESS MODIFICATION

ELAST08

QUIZI

QUIZ OVER ELAST1

ELAST09

ELASTII

INTRODUCTION TO ELASTIC2, CONCEPT OF B-BOX

ADM

BEA AND RELATED IDEAS

PROBII

LOOPING IN ELASTIC2 WITH CODE SEGMENTS

ELAST10

ELAIIB

FORMAL DEFNS OF ENX, INX, LDA, STX, XEJ, XHJ

MACHEQ

MACHINE CODE EQUIVALENTS OF ELASTIC2 INSTRUCTIONS

ELAST11

ELAIII

ARS, QRS, LRS AND CONCEPT OF SHIFTING

LSHIFT

ALS, QLS, LLS AND CODE SEGMENTS

ELAST12

ENTER

ENA, INA, ANA, ORA, COM

MAST

CODE SEGMENTS ON LOGICAL ARITHMETIC

ALIO

DISPLAY CODE, RDA, PRA AND CODE SEGMENTS

ELAST13

LOG

MASKING AND LOGICAL ARITHMETIC

EXTPSD

EXPRESSIONS IN ADDRESS FIELDS

EXTIO

EXTENDED I/O AND PSEUDO-INSTRUCTIONS

ELAST14

QUIZII

FIRST HALF OF QUIZ II

QUIZC

SECOND HALF OF QUIZ II

ELAST15

DIAGN

EXPLANATION OF ERROR DIAGNOSTICS IN ELASTIC

ELAST16

MACROA

INTRODUCTION TO MACROS

MACAA

FORMAT DEFINITIONS CONCERNING MACROS

ELAST17

MACROB

EXAMPLES OF MACROS AND IFT, IFF

MACROC

ERROR DIAGNOSTICS IN MACROS

ELAST18

MACROD

CODE SEGMENTS WITH MACROS

The list of labels that follows was used by students who took the version of the instructional program that was executed on the IBM 1440 computer. Included with this list is the set of instructions that were given to the student for using the list.

CAI ELASTIC COURSE LABELS

The following labels may be used to branch to specific segments of the course that you wish to review. To obtain instruction about a particular subject, you must wait until the green "proceed" light is on, and then you type: go to _______ where the blank is filled in with the appropriate label. The following is a list of labels with a description of the portion of the course to which this label refers.

Label	Description
start	beginning of course
review	decimal, octal, and binary number systems and conversion to and from these bases
begin	ELASTIC - general concepts, machine language
cont	ELASTIC symbolic language
lda	explanation of Load the A register
ldq	explanation of Load the Q register
sta	explanation of Store the A register
stq	explanation of Store the Q register
iad	explanation of Integer Add
isb	explanation of Integer Subtract
unj	explanation of Unconditional Jump
azj	explanation of A Zero Jump
amj	explanation of A Minus Jump
hlt	explanation of Halt
nop	explanation of No Operation
rdi	explanation of Read and Print instructions
imu	explanation of Integer Multiply
idv	explanation of Integer Divide
psedo	explanation of END pseudo-instruction
bss	explanation of BSS Pseudo-instruction
dec	explanation of DEC Pseudo-instruction
dec3	sample ELASTIC 1 program
dec4	exercises in ELASTIC 1
addm	address modification in ELASTIC 1
addm3	quiz #1 - covers ELASTIC 1 and computer concepts

Label	Description
elastic2	beginning of segment on ELASTIC 2
ind9	address modification in ELASTIC 2
ela2ins	explanation of Enter Index
ela2ins2	explanation of Increase Index
ela2ins3	explanation of Load Index
stx2	explanation of Store Index
xhj	explanation of Index High Jump
xej	explanation of Index Equal Jump
xej	numeric word format for ELASTIC 2
elastic3	beginning of segment on ELASTIC 3
shif	explanation of shifting
ars	explanation of A right shift
ars2	explanation of Q right shift
lrs	explanation of Long right shift
als	explanation of A left shift
qls	explanation of Q left shift
lls	explanation of Long left shift
ena	explanation of Enter A register
ina	explanation of Increase A register
ana	explanation of And A register
ora	explanation of Or A register
com	explanation of Complement A register
cdcdc	explanation of CDC 6600 Display Code
rda	explanation of Read Alphabetic
pra	explanation of Print Alphabetic
con	explanation of CON Pseudo-instruction
holl	hollerith constants
mask	masking (logical product)
mask4	logical sum
addf	current instruction address, extensions to DEC, BSS, and END
ib2	extended decimal read
ib5	extended decimal print
ib6	extended octal read and print
ib7	extended alphabetic read
ib8	extended alphabetic print

APPENDIX B

SUPPLEMENTAL COURSE MATERIALS

On the following pages appear the additional handout materials that were used in the classroom to supplement both the lecture and CAI modes of instruction.

C. S. 310

STUDENT DATA SHEET

NAME: ______________________________ CLASSIFICATION: __________

HOMETOWN: __________________________ MAJOR: ________________

I. Please list the courses in computing which you have taken.

Course	Title	Instructor(s)	Grade	Semester Taken

II. Are you taking C. S. 404G concurrently with this course?

III. Please list the computer languages in which you can write programs and the computers for which you have written programs.

IV. Why are you taking this course?

V. What are your long-range occupational goals?

VI. Have you ever had any experience with computer-assisted instruction? If so, please explain.

VII. Please explain the extent of your experience in writing programs in the ELASTIC programming language.

Syllabus for C. S. 310

I. Introduction - present outline of course

II. Number Systems

A. Conversions algorithms
B. Binary
C. Octal
D. Decimal
E. Hexadecimal
F. Binary Coded Decimal
G. Binary Arithmetic
    1. Addition
    2. Subtraction
    3. Multiplication
    4. Division
    5. Complement Subtraction
H. Negative Number Representation
    1. Signed Absolute Value
    2. One's Complement
    3. Two's Complement
I. Exercises

III. Computer Structure

A. Computer Model
B. Memory Organization
C. Computer Logic Design
    1. AND
    2. OR
    3. NOT
    4. Flip-Flops
    5. Half Adder
    6. Adder
    7. Parallel vs. Serial Arithmetic
    8. Encoding and Decoding Logic
D. Registers
E. Information Structures
    1. Bits
    2. Fields
    3. Words
    4. Fixed Word length vs. Variable word length
F. Computer Operation Cycle
G. Characteristics of I/O and Storage Devices
    1. Random Access
    2. Sequential Access
    3. Pseudo-Random Access

IV. Flowcharting
A. Steps in Writing a Program
B. Flowchart Symbols
C. Example Flowcharts
D. Exercises

V. Machine Language
   A. Types of Programming Languages
   B. ELASTIC Instruction Word Format
   C. Principal Instruction Types
   D. Machine Language Programming

Quiz #1

VI. ELASTIC 1 Computer
   A. Character Set
   B. Statement Format
   C. Notation and Terminology
   D. Data Card Format
   E. Instruction Set
      1. Computer Instructions
      2. Pseudo Instructions
   F. Looping
   G. Address Modification
   H. Subroutines
   I. Program Documentation
   J. Error Diagnostics
   K. Programming Exercises

VII. ELASTIC 2 Computer
   A. Index Registers
   B. Numeric Word Format
   C. Instruction Set
   D. Looping
   E. Address and Register Modification
      1. Addressing Techniques
      2. Modification Techniques
   F. Subroutines
   G. Lists and List Processing
   H. Programming Exercises

Quiz #2

VIII. ELASTIC 3 Computer
   A. Shifting
   B. Masking
   C. Instruction Set
   D. Alphabetic Input/Output
   E. Extended Input/Output
   F. Hollerith Constants
   G. Extended Pseudo-Instructions
   H. Programming Exercises

IX. Programming Techniques
   A. Sorting
   B. Searching
   C. Scanning
   D. Symbol Manipulation
   E. Internal Data Conversions
   F. Programming Exercises

X. Macros
  A. ELASTIC Macro Processor
  B. Symbolic Format
  C. Heading, Body and Terminator
  D. Pseudo-Instructions
  E. Nesting
  F. Conditional Assembly
  G. Error Diagnostics
  H. Programming Exercises

Quiz #3

XI. Polish Notation
  A. Conversion Algorithms
  B. Stack Management
  C. List Processing

XII. CDC 6600 Characteristics
  A. System Organization
  B. Central Processor
  C. Peripheral Processor
  D. Data Channels
  E. Multi-Processing
  F. Functional Concurrency

XIII. Representation of Floating Point Numbers
  A. Formats
  B. Conversions and Algorithms

XIV. COMPASS Programming
  A. Computer Instructions
  B. Pseudo-Instructions
  C. Addressing Techniques
    1. Absolute
    2. Relative
    3. Indirect
    4. Indexing
    5. Base Addressing
  D. Subroutines
    1. Organization
    2. Linkage
  E. Macros
  F. Micros
  G. Programming Exercises

XV. Computer Systems Organization
  A. I/O Channels
  B. Interrupts
  C. Time Sharing

XVI. Systems and Utility Programs
  A. Loaders
  B. Monitors
  C. File Manipulation Routines

XVII. Recent Developments

C.S. 310

Spring 1970

SCHEDULE OF CLASS MEETINGS

Date	Explanation
February	
3	Introduction
5	Number systems
10	Computer structure
12	Computer structure
17	Flowcharting and Machine Language
19	Quiz I
24	CAI Instructions, Program Assignments and Documentation
26	NO CLASS*
March	
3	NO CLASS
5	NO CLASS
10	NO CLASS
12	NO CLASS
17	Problem session and quiz review. You should have finished through ELAST10 in the CAI course.
19	Quiz II - Programs as assigned are due (1,2,3,4)
31	Programs assigned and announcement of term project
April	
2	NO CLASS
7	NO CLASS
9	NO CLASS
14	Problem session and quiz review. You should have finished through ELAST18 in the CAI course
16	Quiz III - Programs as assigned are due (5,7,10,11)
21	Review Quiz III, Polish notation and stacks - Evaluation of CAI course due - Turn in all output from CAI course
23	CDC 6600
28	Floating-point representation
30	COMPASS. Programs assigned

*"NO CLASS" applies to the CAI group. The lecture group meets as usual throughout the semester.

Date	Explanation
May	
5	COMPASS
7	Macros, micros
12	System organization and utility programs
14	Recent developments - Programs due - Term project due

C.S. 310 Final Design for Experiment

LECTURE

BATCH	TIME-SHARING
Fr. McNeal	So. Marchak
So. Marshall	Jr. Josue
Jr. Gettman	Jr. Hime
Jr. Raboy	Sr. Peese
Sr. Stanley	Sr. Crow

CAI

BATCH	TIME-SHARING
Fr. Overshiner	Fr. Tilley
So. Hadley	Sp. Bauer
Jr. Aaronson	Jr. Pfiester
Gr. Phien	Gr. Haran
Fr. Phillips	Jr. Bone

BATCH JOB

SUBMISSION PROCEDURES

All jobs to be submitted in batch mode will be placed in one of the input trays in Pearce Hall 105. Your output will be filed according to the job sequence number (orange card) that you placed on the front of your deck. You may retrieve your input cards from the input tray after they have been read through the card reader.

The deck structure for a batch job follows:

N12345.	Job sequence card (orange)
X,,7,77000.CS123456,Name.	Control card (you supply account number and name)
ELASTIC.	Control card to call ELASTIC system
$^{7}8_{9}$	Record separator
*ELAST3	"*ELAST" card - 1st card of program
A BSS 1	
B BSS 1	
C BSS 1	
ST RDI A	
RDI B	ELASTIC program with data cards following the END card
LDA A	
IAD B	
STA C	
PRI C	
HLT	
END ST	
5	
6	
EOF	Green End of File card

CAI EXPERIMENT

TELETYPE

SIGN-UP PROCEDURE

The teletypes that you will use are located in Hogg Building 6. There are a number of smaller rooms that open into this room. The teletypes are in the first small room on the right-hand side after entering the front doors.

The teletype room is locked at all times; therefore, you will have to obtain a key from the Computer Sciences office in Hogg Building 1. You may check the key out for the time you have scheduled on the sign-up sheet which is also located in Hogg B. 1.

Two teletypes have been reserved for the people participating in the CAI experiment at the following times:

10 AM - 12 Noon 1 PM - 5 PM

Monday through Saturday

To reserve a time to use one of the teletypes you should sign the sheet in Hogg B. 1 at least 3 days in advance of the time you desire. Please do not schedule a time more than one week in advance. This will allow everyone an equal chance. Also to facilitate fairness, do not schedule more than a two-hour block of time for a single session at the teletype. (This is about all anyone can take, anyway.)

Your observance of these simple rules will expedite the experiment and give everyone an equal opportunity to participate.

**NOTE: PLEASE KEEP YOUR TTY LOG UP TO DATE!

CAI EXPERIMENT

TELETYPE SCHEDULE

FOR

____

TIME	TTY #1	TTY #2
10:00 - 10:30		
10:30 - 11:00		
11:00 - 11:30		
11:30 - 12 Noon		
12 Noon - 1:00	Not reserved	Not reserved
1:00 - 1:30		
1:30 - 2:00		
2:00 - 2:30		
2:30 - 3:00		
3:00 - 3:30		
3:30 - 4:00		
4:00 - 4:30		
4:30 - 5:00		
5:00 - 10:00	Not reserved	Not reserved

CAI EXPERIMENT

TELETYPE LOG

DATE	SIGN-ON TIME	EXPLANATION OF WORK	TOTAL TIME

DIRECTORY FOR THE PICLS COURSE ELASTIC

USER #	USER NAME	USER TYPE	USER PASSWORD
100	FRED HOMEYER	2	FHOM
101	FG OVERSHINER	2	FOVE
102	LEE AARONSON	2	LAAR
103	TRAN PHIEN	2	TPHI
104	FRED PHILLIPS	2	FPHI
105	MICHAEL HADLEY	2	MHAD
106	RICHARD BAUER	2	RBAU
107	DAVID RAILSBACK	2	DRAI
108	FGP HARAN	2	EHAR
109	DON PFIESTER	2	DPFI
110	CECIL BONE	2	CBON
111	KERN TILLEY	2	KTIL
112	DEMO 1	2	DEM1
113	DEMO 2	2	DEM2
114	DEMO 3	2	DEM3
115	DEMO 4	2	DEM4

ELASTIC ON RESPOND

The necessary information for running ELASTIC programs under the RESPOND time-sharing system is presented below. You should have previously read Bill Alexander's paper, "An Introduction to RESPOND."

Both the control card file for running ELASTIC programs (named ELAST) and the format for entering ELASTIC program statements (named ELAT) are organized in the system as public files available to anyone. Just for reference, however, the control card file is as follows:

```
X,,7,77000.CS123456,NAME.
ELASTIC.
```

Later in the course, if you have to increase the time limit for a job, it will be necessary for you to create your own private control card file. In this file you will need to change the time limit parameter (third parameter from the left on the first record) and also to substitute your account number and your name in the first record.

The public format for entering ELASTIC program statements has tabulation settings to columns 8, 12, and 30. Remember to affect a tab, you must press the control key and tab key at the same time.

LOGGING ONTO RESPOND:

Once you have dialed 9 and then 4766421, determined that RESPOND is active (you hear a steady high-pitched hum on the phone), put the teletype in "REMOTE" or "LINE" mode depending on the model you're using and pressed "DATA" on the data set or put the phone receiver into the acoustic coupler, you are ready to log on.

If RESPOND is active you will receive the following message:

U. T. RESPOND 70

You then type LOGIN, your password and account number and then hit the return key. For example: LOGIN 123ABC CS123456

The RESPOND system will then reply with:

```
TIME  17 00
DATE  02 23 70
PORT  3
....
```

ENTERING A PROGRAM:

The four dots (....) from RESPOND is the signal to you that RESPOND is in the command mode and is awaiting your command. The command mode is used to manipulate files and submit jobs for execution. To build or enter an ELASTIC program, you must get into the data mode. To do this, you should first enter the CLEAR command and then issue an ENTER command as follows:

ENTER ELAT

This command has told RESPOND that you wish to create a file of records using the public format named ELAT. RESPOND then signals you that you are in the data mode by issuing: 00000010=

You may now type in the statement of your program and the data for your program. Remember that the first card of an ELASTIC program must be an "*ELASTI" card where i = 1,2 or 3 depending on which ELASTIC computer you want. Remember also to tab over to the proper fields for entering op codes, operands and comments and to hit the return key at the end of each record. When you have completed entering your program and data you hit the control key and bell key together and then type EXIT. RESPOND replies with .... You then issue a FILE command of the form: FILE progname where progname is a name you wish to associate with the file. You should then issue another CLEAR command.

LOOKING AT A FILE:

To look at your file directory (the list of your private files) you can issue the command FLST FILES or LIST. If you wish to know if a specific file exists you can issue the command: FLST FILES filename where filename is the file of interest. Its existence is signaled by its name being listed by the RESPOND system.

To examine a file you can use the SHOW command or the DISPLAY command. If you just want to look at the file without record numbers you can use:
SHOW filename
to achieve the desired result. To get a listing of record numbers along with the file listing you should use the DISPLAY command. Before using a DISPLAY command; however, the file of interest should be loaded with a LOAD command of the form: LOAD filename
You may then issue: DISPLAY filename
to look at the entire file or: DISPLAY filename, record # TO record #
if you just want to look at specific records in the file.

Always issue a CLEAR command after you have finished looking at your file.

CORRECTING A FILE:

To make corrections on a file, the file must be loaded (by issuing a LOAD command) and you must be in the data mode (issue ENTER ELAT or just ENTER). The procedure described above is performed when you want to insert or replace records in a file.

To delete records from a file, you must load the file and then issue a DELETE command of the form: DELETE filename, record #
or: DELETE filename, record # TO record #.

If you issued ENTER ELAT to get into the data mode you will have to press control and bell and then ij= "text" where ij is a record number and "text" is the new statement to correct or insert a record. Each time you press return, RESPOND will come back with 00000010= . You continue going through the process described above until you have made all insertions and replacements. (A record can have any number even though the format named ELAT initially assigns record numbers in increments of 10.) When all corrections have been made, you hit control and bell and then type EXIT. When you receive .... type a FILE command. It is a good idea when correcting a file to give a new name each time you issue a FILE command. You can then go back and delete all old files when you have obtained a correct version of the file you're working with at the moment.

If you issued simply ENTER to get into the data mode, RESPOND replies with ****. You should supply the record number and = and then type in the text as explained in the other variation of the ENTER command above. The primary difference between these two methods is that with "ENTER ELAT" you can use the tab feature to skip over to specific columns.

EXECUTING A PROGRAM:

To execute an ELASTIC program on RESPOND you use a SUBMIT command of the form:

SUBMIT ELAST INPUT = filename_1 OUTPUT = filename_2

where filename_1 is the name of the file containing your ELASTIC program and data while filename_2 is the name of a file that is to contain the results of program execution.

When your job has been executed you will receive the message: JOB BACK. To look at your results you can issue SHOW filename_2.

To determine the size (number of records) in the file containing results you can issue FLST FILES filename_2. You could then look at particular parts of the file by issuing:

CLEAR
LOAD filename_2
DISPLAY filename_2, record # TO record #

When you have gotten the program to run correctly, you can get a printed listing by issuing: COPY filename_2 TO PRINTER

You pick up the printed listing in Computation Center 18 according to the numeric portion of your password.

DELETING FILES:

Everytime you submit a program for execution or issue a COPY command, you get an additional private file called a dayfile. These files are signified by a "D" after their name when they are listed in the file directory. To remove these dayfiles as well as any other file which you do not need anymore you issue a DELETE command of the form: DELETE FILE filename

LOGOUT and SAVE:

To log off the system you issue: LOGOUT You should then turn the teletype off and hang up the phone.

If you find yourself in the middle of a long session and you wish to protect or safeguard the work you have already done in the session you can issue a SAVE command. This command causes an automatic log out and log on again. The form of the command is: SAVE

STOPPING A LONG LISTING

If you don't want to look at the entire listing of a file after you have issued a SHOW or DISPLAY command, you can hit control and bell and then type in STOP. RESPOND will reply with: ....

CORRECTING A COMMAND OR RECORD PRIOR TO ISSUANCE:

To correct typographical errors in a command or record prior to hitting the return key, you use a combination of the shift key and ← . The character ← serves to delete itself and the previous character.

To cancel a RESPOND command before issuing it you hit control and bell and then hit the return key.

NOTES OF SIGNIFICANCE:

1. Please keep all teletype output throughout the semester and turn it in on the last day of class.

2. Please keep your teletype log up to date and as detailed as necessary for determining what you accomplish during each session at the teletype.

3. Schedule your teletype time in advance by signing the schedule sheet in Hogg Building 1.

4. If you have trouble with the teletype, the consultants in the room immediately to the left of the teletype room will help you.

5. Mark places on your teletype output where you encountered difficulty.

EXAMPLE SESSION ON RESPOND:

EXAMPLE:

(ALL UNDERLINED LINES DENOTE USER-ENTERED LINES)

∇ denotes control bell

```
U.T.RESPOND . 70
LOGIN 180HJD CS290280
 TIME 14 40
 DATE 18 02 70
 PORT  1
.... FLST FILES
PRIVATE   FILES
ELAST           2 DIS
DISK SPACE ASSIGNED    40,USED       3
.... ENTER ELAT
00000010=A       BSS 1
00000020=B       BSS 1
00000030=STRT    RDI A
00000040=        RDQ B
00000050=        LDA A
00000060=        IAD B
00000070=        STA SUM
00000080=        PRI SUN←M
00000090=        HLT
00000100=        END STRT
00000110=2
00000120=5
00000130=7
00000140=∇EXIT
.... FILE PROG
.... FLST FILES
PRIVATE   FILES
ELAST           2 DIS
PROG           13 DIS
DISK SPACE ASSIGNED    40,USED       6
.... CLEAR
.... SHOW PROG
A       BSS 1
B       BSS 1
STRT    RDI A
        RDQ B
        LDA A
        IAD B
        STA SUM
        PRI SUM
        HLT
        END STRT
2
5
7
.... CLEAR
.... LOAD PROG
```

```
.... DISPLAY FILE PROG 20 TO 40

00000020 =B       BSS 1
00000030 =STRT    RDI A
00000040 =        RDQ B
.... DISPLAY FILE PROG 30
00000030 =STRT    RDI A
.... ENTER ELAT
00000010=∇9=*ELAST1
00000010=∇21=SUM    BSS 1
00000010=∇40=       RDI B
00000010=∇EXIT
.... FILE NEW
.... CLEAR
.... LOAD NEW
.... DISPLAY FILE NEW
00000009 =*ELAST1
00000010 =A       BSS 1
00000020 =B       BSS 1
00000021 =SUM     BSS 1
00000030 = STRT   RDI A
00000040 =        RDI B
00000050 =        LDA A
00000060 =        IAD B
00000070 =        STA SUM
00000080 =        PRI SUM
00000090 =        HLT
00000100 =        END STRT
00000110 =2
00000120 =5
00000130 =7

.... CLEAR
COMMAND UNRECOGNIZED
.... CLEAR
.... LOAD NEW
.... DISP FILE NEW 130
00000130 =7
.... DELETE FILE NEW 130
.... CLEAR
.... FLST FILES
PRIVATE    FILES
ELAST          2 DIS
PROG          13 DIS
NEW           15 DIS
DISK SPACE ASSIGNED    40,USED       9
.... LOAD NEW
.... DISPLAY FILE NEW 130
00000130 =7
.... DELETE FILE NEW 130
.... FILE NEW
.... FLST FILES NEW
```

```
PRIVATE   FILES
NEW            14 DIS
DISK SPACE ASSIGNED    40,USED      9
.... DELE FILE PROG
.... CLEAR
.... LOAD NEW
.... FILE NEW E←FORT
.... CLEAR
.... SUBMIT ELAST INPUT=NEW OUTPUT=RESULT
SYSTEM JOB NAME    X180NFX
.... STATUS X180NFX
JOB IN OUTPUT
.... JOB BACK
FLST FILES
PRIVATE   FILES
ELAST           2 DIS
NEW            14 DIS
ELAST  D        8 DIS
RESULT         39 DIS
DISK SPACE ASSIGNED    40,USED      12

CLEAR
.... SHOW ELAST D
118 FEB 70   UNIVERSITY OF TEXAS 6600   UT 1
 14.48.37. X180NFX. **RESPOND**
 14.48.41. X180NFX. X,,7,77000.CS290280,HOMEYER.
 14.48.43. X180NFX..ELASTIC.
 14.48.43. X180NFX.   EXIT
 14.48.43. X180NFX. CP  000.310 SEC.
 14.48.43. X180NFX. PP  001.084 SEC.
 14.48.43. X180NFX. TM  001.278 SEC.       2 (OCTAL)
.... DELETE FILE ELAST D
.... SAVE
 TIME 14 50
 DATE 18 02 78
 PORT  1
.... CLEAR
.... LOAD NEW
.... DISPLAY FILE NEW
00000010 =*ELAST1
00000020 =A       BSS 1
00000030 =B       BSS 1
00000040 =SUM     BSS 1
00000050 =STRT    RDI A
00000060 =        RDI B▽ STOP
.... CLEAR
```

```
.... SHOW RESULT
1*************************************************************************
**************************************************************'
0
0
0
          ELAST1

0
 BEGIN RUN

          0000   000000000000     A      BSS 1
          0001   000000000000     B      BSS 1
          0002   000000000000     SUM    BSS 1
          0003   650000000000     STRT   RDI A
          0004   650000000001            RDI B
          0005   120000000000            LDA A
          0006   140000000001            IAD B
          0007   200000000002            STA SUM
          0010   660000000002            PRI SUM
          0011   000000000000            HLT
                                         END STRT
OMULTIPLE-DEFINED SYMBOL. ADDRESS USED.

        *NONE*

        UNDEFINED SYMBOL. ADDRESS USED.

        *NONE*
1
 ASSEMBLY COMPLETE.
 NUMBER OF INSTRUCTIONS PROCESSED -     11
 STARTING LOCN 0003
 ASSEMBLY TIME -     .233 DECIMAL SECONDS
 (0002) = DECIMAL              7
 HLT 000000000000 ENCOUNTERED AT LC = 0011. EXECUTION STOPPED.
 NUMBER OF INSTRUCTIONS EXECUTED -      7
 EXECUTION TIME -     .012 DECIMAL SECONDS
.... COPY RESULT TO PRINTER
SYSTEM JOB NAME     X180PKX
.... JOB BACK
FLST FILES
PRIVATE   FILES
ELAST          2 DIS
NEW           14 DIS
RESULT        39 DIS
RESULT D       7 DIS
DISK SPACE ASSIGNED    40,USED      12
.... DELE FILE RESULT D FILE RESULT FILE NEW
.... FLST FILES
PRIVATE FILES
ELAST         2 DIS
DISK SPACE ASSIGNED    40,USED        3
```

```
.... LIST PUBLIC FILES
PUBLICP   FILES
LGOCC          2 DIS    VL      2/18/70        P3
RUNGO          3 DIS    VL      2/18/70         3
RUNT           3 DIS    VL      2/18/70         3
XRUNGO        26 DIS    VL      2/18/70         3
COM       *    3 DIS    VL      1/20/70         3
DES            3 DIS    VL      2/18/70         3
TEX            3 DIS    VL      2/11/70         3
LCF          355 BIN    20      2/1/70         P2
XLCF           9 DIS    VL      2/12/70         3
LIS            4 DIS    VL      2/18/70         3
DESCAL       921 DIS    VL      2/18/70        20
ALGCC          3 DIS    VL      2/ 4/70         3
SCOOP         19 DIS    VL      2/18/70         3
EXPIRE       279 DIS    VL      2/1X/70         7
SKED          26 DIS    VL      2/18/70         3
DISK SPACE ASSIGNED    40,USED        3
.... LIST PUBLIC FORMATS
PUBLIC    FORMATS
CCF  RECORD DIS    VL 00000010  NO  0010
FORT RECORD DIS    80 00000010  NO  0010 TABS   7
.... LIST
PRIVATE   FILES
ELAST          2 DIS    VL      2/18/70         3
DISK SPACE ASSIGNED    40,USED        3
PRIVATE FORMATS
ELAT RECORD DIS    VL 00000010  NO  0010 TABS   8,  12,  30
.... SHOW SCOOP
JANUARY 29, 1970.
**** RESPOND USERS *********
EFFECTIVE AT 0800 FEB 2, 1=70 NEW PROBLEM NUMBERS WILL BE IN EFFECT.


ANY RESPOND FILES NOT ACTIVATED SINCE JAN 1, 1970 WILL BE PURGED
ON FEB 1, 1970.

JANUARY 22,1970
  A RESTRICTION HAS BEEN ADDED TO THE CONVERSE FEATURE.
THIS RESTRICTION IS ONE THAT PROHIBITS THE USE OF THE
CONVERSE FEATURE WITH AN EQUIPMENT ASSIGNED.  IN PARTICULAR,
USE OF CONVERSE WITH A TAPE ASSIGNED TO THE JOB IS NOT
PERMITTED.  CONVERSE JOBS WHICH NEED A TAPE FOR DATA
INPUT HAVE TO COPY THE TAPE TO A DISK FILE AND RETURN THE TAPE
PRIOR TO EXECUTING THE CONVERSATIONAL PORTION OF THE JOB.
SIMILARLY, JOBS WHICH NEED TO SAVE DATA ON A TAPE MUST ACCUMULATE
THEIR DATA ON A DISK FILE DURING THE CONVERSATIONAL PORTION
OF THEIR JOB.  AFTER COMPLETING THIS JOB STEP, THEY MAY
REQUEST A TAPE AND COPY THE DISK FILE TO THE TAPE.
```

.... SHOW SKED

STARTING MAY 26, 1969   THE RESPOND SCHEDULE IS
MONDAY-FRIDAY   0930 A.M. TO 0330 A.M.
SATURDAY   0930 A.M. TO MIDNIGHT
****

NOTE: ON THE FIRST AND THIRD MONDAYS OF EACH
MONTH RESPOND WILL GO OFF THE AIR AT 2330.

****
THESE DUMPS CAUSE APPROX. A 25 MIN. DELAY, DURING
WHICH RESPOND IS INACTIVE. OTHER DUMPS WILL BE TAKEN
IN THE EVENT OF A SYSTEM FAILURE.
PLEASE NOTICE THAT STARTING TIMES ON ANY SCHEDULED
PERIOD ARE APPROXIMATE START UP TIMES. ACTUAL START UP
TIME DEPENDS UPON THE LENGTH OF TIME REQUIRED FOR LOADING.
THE TRAY FOR RESPOND COMMON FILES WILL BE LOADED
ONCE PER DAY, AT 0930. USERS NOT GETTING THEIR
JOBS INTO THAT TRAY MAY, OF COURSE, USE THE REGULAR
INPUT TRAY FOR THE JOB. IN THE LATTER CASE,
THERE IS NO GUARANTEE THAT THE COMMON FILE WILL
GET IN AT ANY PARTICULAR TIME.
.... LOGOUT
TOTAL*00.10.59

INFORMATION

ON

PICLS

As a student taking the ELASTIC computer-assisted instruction course you will be using facilities of both the RESPOND time-sharing system and the PICLS instructional system. This sheet presents the information that you need concerning PICLS.

You have two PICLS commands at your disposal. A command to PICLS is signified by preceding it with a $. The commands available to you are as follows:

1. $LESSON, section name

where section name is the name of the section you wish to take. If you are signing onto a particular subcourse for the first time, you will want to issue the command: $LESSON,START in order to get a list of the sections contained in the subcourse. The LESSON command initiates execution of a section.

2. $ENDRUN

You should issue this command when you are ready to terminate a session with the CAI course. The PICLS system will always respond with "DO YOU WISH TO SAVE YOUR FILES." You should always respond with NO.

**NOTE: Every time you finish a section, PICLS will ask "DO YOU WISH TO GO ON TO THE NEXT SECTION?" If you say YES you will be automatically sequenced to the next section. If you say NO, PICLS will then say, "YOU HAVE COMPLETED LESSON section name NOW TYPE "$LOGOFF" OR $LESSON" TO EXECUTE A NEW LESSON. At this point you may type $LESSON, section name or $ENDRUN. Never enter $LOGOFF.

FACTS AND PROCEDURES ON PICLS

This sheet will present some facts and procedures concerning the use of PICLS and the CAI course on ELASTIC.

***1. As of February 20, 1970, the procedure for obtaining a subcourse of the CAI course is as follows:

CONVERSE PICLS INPUT=ELASij CAIDIR

where i and j are numbers designating which subcourse you desire. Notice that the T is left off of ELAST in the filename above. This is due to naming conventions on RESPOND public files.

***2. When beginning a subcourse for the first time, type $LESSON,START to receive a list of the sections within this subcourse. You will then be automatically sequenced into the first section.

***3. If you should get the message "JOB BACK" in the middle of a CAI session, this means you have run out of allotted system time. Log back into the PICLS system (by issuing a CONVERSE command) and type $LESSON,name where name is the next section in sequence after the one on which you ran out of time. When it is convenient for you, notify Mr. Homeyer about the fact that you ran out of time.

***4. You will be notified by the system, every time you have completed a section. To obtain the name of the section just completed, respond NO when asked if you wish to go on. You should then type $LESSON,name where name is the name of the next section in sequence when you receive the * again from the system.

***5. <u>ALWAYS WAIT FOR THE * BEFORE TYPING ANY CHARACTERS!!!!!!</u> If you forget, you may have to log onto PICLS again (by using a CONVERSE command).

***6. When you are notified that you have reached the end of a subcourse you should always respond with NO when asked if you wish to go on. Enter $ENDRUN and then NO to logoff the system. Log back on again with a CONVERSE command.

***7. If the RESPOND system goes down (signified by "U. T. RESPOND" being typed in the middle of a session), log back onto RESPOND (using a LOGIN command) and then log back onto PICLS and begin at the start of the section you were taking.

***8. You can log off of PICLS any time you receive an * by entering $ENDRUN. However, you can only re-enter the CAI course at the beginning of a section.

***9. You can stop the text output in the middle of a section by hitting <u>control</u> and <u>bell</u> and then entering STOP. You will receive an * at which time you should log off of the system by entering $ENDRUN.

***10. When taking the CAI course, always wait for the *, enter your response and then signal the end of your response by hitting the <u>return</u> key.

***11. Save all of your teletype output and turn it in at the end of the semester along with your teletype log sheet.

***12. The consultants in Pearce Hall 105 will help you with any programming and debugging problems you have running ELASTIC programs.

***13. You can execute an ELASTIC program on-line during a CAI section by logging off of PICLS and then building a program file and executing it under RESPOND by using a SUBMIT command. TRY TO USE THIS ON-LINE EXECUTION FEATURE AS MUCH AS POSSIBLE, WHEN TAKING THE CAI COURSE.

APPENDIX C

TESTS AND PROGRAM ASSIGNMENTS

Copies of the tests and program assignments given to both the lecture and CAI groups are presented in the following pages. The documentation form which was used as a guide for program writeup is also included.

C.S. 310

DEPARTMENT OF COMPUTER SCIENCES

Spring, 1970, Quiz #1

I. Base Conversions: Perform the indicated conversion in each of the following problems.

1. 7340 (eight) = __________(ten)
2. 51.078125 (ten) = ________(eight)
3. 43.125 (ten = __________(two)
4. 110110111.110110111 (two) = ____________(ten)
5. 7564012.346 (eight) = ______________________________(two)
6. 11011100000101.01101 (two) = ____________(eight)
7. 4CDF.AB (sixteen) = ________________________________(two)
8. 1011010111101011111.1011011110111 (two) =__________________(sixteen)
9. 6942.365 (ten) = ________________ (BCD)
10. 100110000111.0000001001 (BCD) = ______(ten)
11. 221010 (three) = ____________(five)
12. 110111.101 (ten) = ___________________(two)

II. Binary Arithmetic: Perform the indicated operation in each problem.

(1)
```
  101011.11
+   1101.11
```

(2)
```
  101110.101
-   1101.11
```

(3)
```
  10111111
X     1001
```

(4) 10.1   11101010.10001

(5) complement subtraction
```
   101101
-10111011
```

(6) complement subtraction
```
 101101.1
-110101.01
```

(7) complement subtraction
```
      11.01
-1101101.110
```

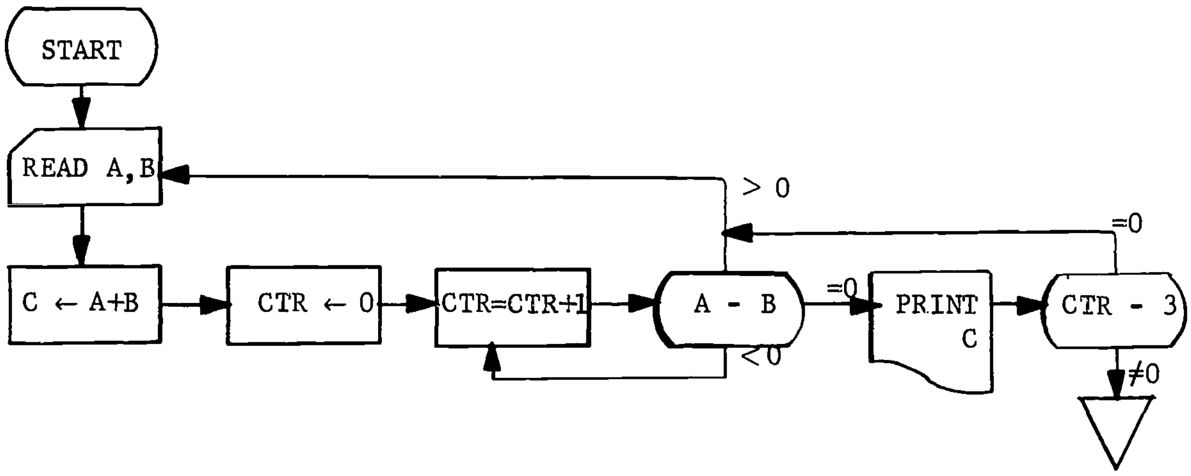
(8) complement subtraction
```
 10100111
- 1101011
```

Page 2

III. Definitions: Give a brief definition of each of the following terms.

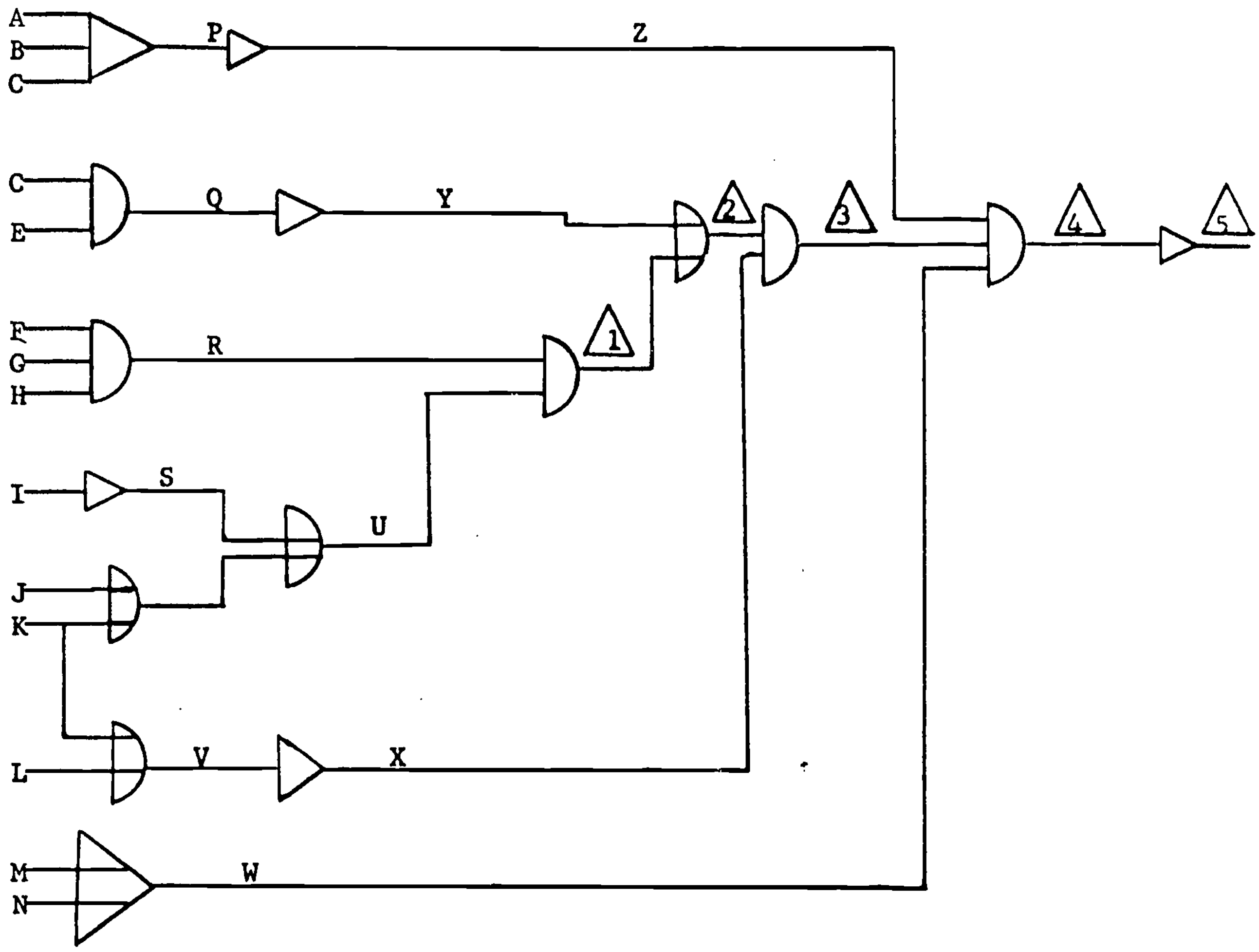
1. BCD
2. assembler
3. CAI
4. bi-stable
5. synthetic substitution
6. BIT
7. synchronous
8. ELASTIC
9. central processing unit
10. inhibit wire

IV. <u>Flowchart</u>: Study the flowchart below and then answer the questions posed.



1. How many times will point X be reached in this flowchart? __________
2. How many times will C be printed? __________
3. How many data cards will be read? __________

V. <u>Circuitry</u>: After studying the circuit network presented answer the questions below.



If △5 = 0, △1 = 0, R = 1, and D = 1, then A = ____, B = ____, C = ____, D = ____,

E = ____, F = ____, G = ____, H = ____, I = ____, J = ____, K = ____, L = ____,

M = ____, N = ____, P = ____, Q = ____, S = ____, T = ____, V = ____, W = ____,

X = ____, U = ____, Y = ____. Z = ____ △2 = ____, △3 = ____, and △4 = ____.

Page 4

VI. True-False: If the statement as written is true, write "true" in the blank to the left of the statement. If the statement is false, write "false" in the blank to the left of the statement and <u>state under the statement, what change is required to make the statement true in a sense relative to the subject discussed</u>.

_______1. A light bulb is an example of a tri-stable device.

_______2. Each storage location in the memory of a computer has a unique address associated with it.

_______3. After reading the state of a ferrite core, the core is restored to its original state.

_______4. Most scientific computers use a variable word length.

_______5. When performing complement subtractions, if you get a zero in the overflow position then you should prefix your answer with a minus sign.

_______6. The circuitry in a computer shapes and times electronic pulses, stores bits and performs logic.

_______7. To compute the 5s complement of a given base six number, subtract each of the given numbers from 10 (base six).

_______8. Internal memory holds numbers which are data or information.

_______9. The five basic elements of any computer system are the overflow unit, the instruction register, the input unit, the memory unit, and the control unit.

______10. In a variable word length computer, the operation of the circuits which perform arithmetic may be described as being serial in nature.

______11. Every computer instruction must go through an interpretation phase and a decoding phase before entering the execution cycle.

______12. The BCD number system sacrifices efficiency for convenience.

______13. Most "business" computers are concerned with applications that are computation-bound.

______14. Computer instructions specify the execution of small, well-defined steps or operations.

______15. Every computer instruction must be comprised of an operation code and an operation.

______16. The read cycle for sensing the state of a ferrite core consists of writing a 1, testing the sense wire, and then trying to write a 0.

Page 5

_______17. If a computer application is input/output bound then a fixed-word-length computer should be used.

_______18. The largest number than can be represented by 52 bits is $2^{31} - 1$.

_______19. Precision refers to the number of significant digits used to express a number.

_______20. A computer word is a set of characters treated as a unit by the computer.

_______21. The time to perform a computer operation is the sum of access time, time sharing, real time, time to translate the operation code and execution time.

_______22. The memory unit is the "central nervous system" of the computer.

_______23. The control unit is the "heart" of a computer system.

_______24. The fact that 32 (five) equals 17 (ten) is well known.

_______25. A computer can solve a problem entirely by itself.

_______26. An example of an output device is magnetic tape.

_______27. Memory addresses are assigned sequentially in the memory unit beginning with 1.

_______28. The use of a computer enhances the need for a thorough understanding of the problem area.

_______29. A program can be thought of as "a schedule of events."

_______30. A flowchart is concerned with illustrating the program logic and processing sequence.

_______31. Testing a program to eliminate errors is called debugging.

_______32. A computer can be thought of as an extremely fast adding machine.

_______33. There are two basic types of computers, analog and hybrid.

_______34. A pencil is an example of an analog device.

_______35. To build an adding machine that does a base three arithmetic, each counting device in the machine must be able to represent the digits 1 through 3.

_______36. A nanosecond is one-millionth of a second.

_______37. The speed with which electricity passes through computer circuitry is limited by the speed of light.

Page 6

_______38. Ferrite devices are chosen as memory components because they will not lose their state when the power is removed from the system.

_______39. Machine language is one of many languages which the computer understands.

_______40. Assembly language is less detailed than procedure-oriented language.

_______41. The primary purpose of the input unit is to accept information in human-readable form and translate the information into two-state form for storage in the control unit.

_______42. Most modern-day computers utilize the stored-program concept.

_______43. Conversion from decimal to binary may be accomplished using the synthetic substitution method.

_______44. The word computer is derived from the Latin word "computare."

_______45. In a fixed-word-length computer, parallel data transmission may be realized.

_______46. Speed and accuracy are two distinct attributes of the computer.

_______47. The computer is a giant brain that knows all things.

_______48. All information intended for computer consumption must be presented to it in two-state form.

_______49. The most efficient number system for internal representation in a computer is base three.

_______50. Each word in memory has a unique address associated with it.

C.S. 310

Spring, 1970, Quiz #2

I. Discussion and True-False: Fill in the blank, answer true or false, or give a brief discussion for each of the following questions.

1. The total range of addresses available for programming use in the ELASTIC 1 computer as represented in octal is _______ to ______.

2. Name two operational registers in the ELASTIC 1 computer.
   1. ______________________ 2. ___________________

3. Some instructions in ELASTIC 1 treat the combined QA register as one 72-bit register. (true-false) ________

4. An assembler program or routine is a computer program which ____________
____
____

5. Instructions must be written in symbolic format for input to the ELASTIC 1 assembler program. (true-false) ________

6. Give a brief but concise definition of a subroutine.

7. List three advantages for writing a program as a group of subroutines instead of one large program.

8. Why is program documentation important?

9. A ___________ subroutine usually appears in-line in the program code each time it is used while a ___________ subroutine is coded only once and then called through a calling sequence in the main program.

10. Numbers in computer memory are either ___________ or ___________.

11. Give the symbolic format for an ELASTIC 1 instruction.

12. Give the numeric format for an ELASTIC 1 instruction.

13. If a symbol must begin in the leftmost column of a field, then the symbol is said to be ________________ in the field.

14. A field is ____

15. The two types of instructions in ELASTIC 1 are ___________ instructions and ______________ instructions.

16. To what does the notation "(...)" refer?

17. Why are assembly languages termed machine dependent?

18. Why were assembly languages developed?

19. Before the IMU instruction may be executed it is necessary that the multiplicand be loaded in the A register. (true-false) ___________

20. To achieve proper execution of the IDV instruction it is necessary that the Q register contain positive or negative zero depending upon the sign of the dividend. (true-false) _________

II. Give the indicated answer for each question, utilizing the following information:

```
(TEM)   = 000000000043
(WORD)  = 000000000007
(CELL)  = 000000000014
(ONE)   = 000000000001
(MZERO) = 777777777777
```

1. If (Q register) = 000000000000 and (A register) = 000000000062 and the following instruction is given:

```
IDV  WORD
```

what is the final contents of the of the A register? Q register? WORD?

(A register) = ______________ (Q register) = ______________

(WORD) = ______________

2. If (Q register) = 000000000727 and the following instruction segment is given:

```
LDA  TEM
IMU  CELL
```

what is the final contents of the A register, Q register, and TEM?

(A register) = ___________ (Q register) = ___________ (TEM) = _________

3. Given the following code segment:

```
      LDA  WORD
      ISB  CELL
      AMJ  JUMP
      LDQ  TEM
JUMP  LDQ  ONE
```

what is the final contents of the A register and the Q register?

(A register) = _______________ (Q register) = ___________________

4. Given the following code segment:

```
        LDQ  TEM
        LDA  MZERO
        AZJ  EXIT
        AMJ  JUMP
        UNJ  SKIP
        LDQ  ONE
JUMP    LDA  WORD
EXIT    IAD  ONE
SKIP    IAD  ONE
```

give the final contents of the A and Q registers.

(A register) = ________________ (Q register) = ________________

5. Using only (WORD) make the final contents of WORD equal to its original contents plus 1.

6. Using only (ONE) make the final contents of ONE equal to positive zero.

7. Given the following sequence of code, what does the word named OP contain after two iterations of the loop?

```
ARRAY   BSS  2
CTR     DEC  0
ONE     DEC  1
LIM     DEC  6
OP      RDI  ARRAY
        LDA  OP
        IAD  ONE
        STA  OP        LOOP FOR READING
        LDA  CTR
        IAD  ONE
        ISB  LIM
        AMJ  OP
```

8. How many times will the loop in question 7 be executed?

9. What, if anything, is wrong with the following calling sequence to a subroutine named SUB?

```
        LDA  ADR
        LDQ  RTN
        UNJ  SUB
ADR     HLT  DATA
RTN     UNJ  NEX
NX
```

10. Program Exercise: Write a program which will read N decimal values from cards and then compute the average of the N values. Print the input values after you have read them and print the average after you have computed it. Write your program with enough generality so that you could handle any number of values. Use address modification to read and print the input values.

Spring, 1970, Quiz #3

I. Discussion, True-False: Give a brief but concise answer or answer true or false to the following questions:

1. How many index registers are available in ELASTIC?

2. The contents of B0 is always what?

3. What is another name for an index-register?

4. In the instruction LDA WORD+6 (4), what is the underlined portion called?

5. What type of constants are created by the variation of the CON pseudo-instruction in which an "H" appears in column 12 of the card?

6. Name the two general categories of arguments that may be transmitted to a subroutine.

7. List three capabilities in ELASTIC3 that were not available in ELASTIC1.

8. The instruction INA 5, when executed, causes the contents of the cell whose address is 0005 to be entered into the A register. (true-false)

9. The instruction ENX 377B(4), when executed causes the constant 377B to be entered into B4. (true-false)

10. The instruction INA CELL, when executed, increases the contents of the A register by the address of CELL. (true-false)

11. In the ELASTIC computer, all left shifts are performed end-off. (true-false)

12. The instruction LDX TEMP(4), when executed, loads index register 4 with the address of TEMP. (true-false)

13. The instruction STX WORD(6), when executed, causes the contents of WORD to be stored in index register 6. (true-false)

14. The instruction ENX DATA(1), when executed, causes (DATA) to be entered into B1. (true-false).

15. All right shifts in ELASTIC are performed end-off. (true-false)

II. Discussion: Provide the information requested.

1. Write a program segment to add the contents of index registers B1, B2, B3 without using any auxiliary label names, into the A register.

2. Place in the A register the negative of the contents of index register 4, without using any labels.

Page 2, Quiz #3

3. Place in the Q register the contents of B4, without using any labels.

4. Write an instruction which will exchange the current contents of the A and Q registers.

5. Give the general form for the octal format of an ELASTIC3 instruction.

6. Write an instruction which will cause a transfer of control to a word named JUMP if (B3) is greater than 23 octal.

7. Given that (B4)=0004, how many times will the loop be executed?

```
ENX   0(2)
LDA   G(2)
IAD   0(2)
STA   0(2)
INX   1(2)
XEJ   $ -4(4),6
```

8. Give the contents of ST after executing the code segment presented. (The numeric op code for ENA is 10 and for INA is 11.) Assume address of ST is 1000

```
ST   ENA     0
     LLS    36
     INA    10
     INA   $ -3
     INA   $ -2
     COM
     STA   ST
```

(ST) = ____________________

9. Write a code segment that creates and prints the characters: TEXAS LONGHORNS

10. Write an instruction which will transfer control unconditionally to a location 7 after itself.

III. Assume you are given the sentence below punched on a data card starting in card column 1:

THAT BRIDGE IS A BRIGHT ONE

1. Write a program segment to read this sentence into an array named DATA. Contract the sentence to:

THAT BRIDGE IS BRIGHT

Store the contracted sentence in an array named SMALL and then print the contracted sentence.

Page 3, Quiz #3

2. Given the following program segment and the input sentence from question 1, show a copy of the output.

```
DATA    BSS    10
MASK    CON    777777777700B,1
        ENX    0(2)
        RDA    DATA,DATA+4
LOAD    LDA    DATA(2)
        ANA    MASK
        STA    DATA(2)
        INX    1(2)
        XEJ    CAT(2),3
        UNJ    LOAD
CAT     PFA    DATA,DATA+3
        HLT
```

3. If the instruction ANA MASK is replaced by ORA MASK+1, show a copy of the output. (You may use the list of display codes from the manual if you desire.)

DEPARTMENT OF COMPUTER SCIENCES

C.S. 310

Final Examination-Spring 70

Instructions: Write all answers on the answer sheet that is provided with your copy of the examination. If the statement in a particular problem is true, write the letters "TRUE" in the blank on the answer sheet corresponding to the number of the problem involved. If the statement is false, write the letters "FALSE" in the blanks on the answer sheet. If an answer of 0 or 1 is asked for in a particular problem, write this value on the blank on the answer sheet.

I. For this section write true or false in the blank on the answer sheet corresponding to each problem.

1. (729.4)eight = (1110101001.100)two
2. (457.625)ten = (711.5)eight
3. (641.5)eight = (111100001.101)two
4. (534.1)eight = (348.125)ten
5. (111101110001.110011)two = (7651.63)eight
6. (1011011.1)two = (91.5)ten
7. 1101101101 + 1101101011 = 11011011000 (binary arithmetic)
8. 11011 x 100 = 11011000 (binary arithmetic)
9. 11010 - 11 = 10001 (binary arithmetic)
10. The one's complement of 11100111 is 00011000.
11. The seven's complement of 20043765325 is 47734012452
12. The number 200240000000 represents a number in floating-point format. This number represented in fixed-point octal form is 4.
13. The number 577377777777 represents a number in floating-point format. This number in fixed-point binary form is $2^{47}-1$.
14. The number 1775400000000000 represents a number in floating-point format. This number represented in fixed-point octal form is 0.1.
15. The number 600537777777 represents a number in floating-point format. This number represented in fixed-point octal form is +0.04.
16. The floating-point representation for the number 40.08 is 200640000000.
17. The floating-point representation for the number -48 is 577377777777.
18. Instructions must be written in octal format for input to the ELASTIC assembler program.
19. In some instructions it is possible to operate on 72 bits as though they made up the contents of one register.

Page 2, Final Examination

20. If the contents of the A register are positive, the instruction AMJ JUMP will cause a transfer of control to location JUMP.

21. If (Q register) = 000000000000, (A register) = 000000000027 and (DIV) = 000000000003 and the following instruction is given: IDV DIV, then after execution of this instruction (Q) = 000000000002, (A) = 000000000007 and (DIV) = 000000000000.

22. Before the IMJ instruction is referenced, it is necessary to load the multiplier into the A register with a previous instruction.

23. Before the IDV instruction is referenced, it is necessary to have a positive or negative zero in the Q register depending on the sign of the divisor.

24. If (A register) = 000000000027, (TEMP) = 000000000002 and (Q register) = 0000000000036 and the following instruction is referenced: IMU TEMP, then the final (Q) = 000000000000, (A) = 000000000056 and (TEMP) = 000000000002.

25. When a READ instruction is executed, the contents of the card which is read, is stored in the A register.

26. If (B5) = 0002 and the following instruction is given: LDA TEMP+14(5), the contents of the cell located 10 positions after location TEMP in memory is put into the A register.

27. If (A register) = 000000000143 and (CELL) = 000000000143 and the following instruction is given: ISB CELL, then the final (A register) is 000000000012, and (CELL) = 000000000014.

28. If (A-register) = 777777777777 and the instruction AZJ SKIP is given, control transfers to location SKIP.

29. If (A-register) = 000000000012 and (TEMP) = 000000000002 and the instruction IMU TEMP is given, the final (A-register) is 000000000024.

30. The instruction ARS 5 would cause the (A register) to be shifted five places to the right. The five bits shifted off the right end of the A register appear in the high order five bits of the A register after execution of the instruction.

31. If (B6) = 0004 and the address of STORE is 1526, then the instruction STA STORE+2(6) will cause the present contents of the A register to be placed in storage location 1534.

32. If (B2) = 7774 and the address of CELL is 0014, then the instruction STQ CELL(2) will cause the present contents of the Q register to be laced in storage location 0011.

33. If an index register is designated in the IAD instruction, the contents of the designated index register are added to the previous contents of the A register.

Page 3, Final Examination

34. If (B3) = 0014 and the instruction XEJ OUT(3), 14 is encountered, control passes to the location named OUT.

35. The instruction ENX CTR(2) causes the contents of the word named CTR to be placed into B2.

36. When performing complement subtraction, if you get a zero in the overflow position, then you should recomplement your answer and leave the sign alone.

37. To write or represent a one on a ferrite core, the first operation is to read from the core in order to sense its state.

38. The ability to represent the absolute value of a number using a multi-state device depends on the device's having different recognizable states.

39. Internal memory can hold only numbers which are data.

40. A memory device using the polarity principle will lose its polarity as soon as the power is turned off.

41. Precision refers to the number of significant digits which can be used to express a quantity.

42. One of the difficulties of symbolic language as opposed to absolute machine language is the fact that it is difficult to change a program at a later date.

43. Error diagnostics with respect to correct coding of operation codes are not usually included in an assembled listing of a symbolic program.

44. A program written in symbolic language is called an object program before it is assembled.

45. Good flowcharting techniques are an important factor in developing operational programs.

46. The largest number that can be represented by 18 binary digits is 2 raised to the 18 power.

47. Computer instructions specify execution of small, well-defined steps or operations.

48. An assembler program corrects a source program's semantics.

49. A flip-flop is a bistable device capable of assuming two stable states.

50. The use of a computer reduces the need for a full understanding of the problem area.

51. Including B0, there are eight index registers in the ELASTIC computer.

52. Two types of arguments which can be transmitted to a subroutine are values and address.

53. The instruction LDX 1777B(1), when executed causes the constant 1777B to be loaded into index register 1.

54. The instruction ENX 3777B(2), when executed, causes the constant 3777B to be entered into B2.

55. In the instruction IMU MJL+2(4), the underlined portion is called the effective address.

56. The instruction INA TP, when executed, increases the A register by the address of TP.

57. The instruction ENA 5, when executed, causes the constant 5 to be entered into the A register.

58. The instructions ARS, QRS, and LRS cause the contents of the register(s) involved to be shifted right, end-around the number of positions specified by the sum of the BEA and the contents of the designated index register.

59. If the instruction COM is given, this causes the sign bit of the number in the A register to be reversed and the remainder of the number is left unchanged.

60. If (B2) = 0014 and the instruction XEJ OUT(2), 14 is given, control passes to the location named OUT.

61. If (B4) = 7772 and the instruction ENX 122(4) is given, the final contents of B4 is 0122.

62. If the address of the location named X is 0010 and (B3) = 0007, then the instruction ALS X(3) causes the contents of the A register to be shifted left end-around 15_{ten} bits.

63. The instruction ENA 0(6) will in effect place the contents of B6 in the A register.

64. The instruction INA SUM causes the contents of the A register to be increased by the contents of the cell named SUM.

65. The instruction ANA D forms the logical sum of the contents of the A register and the contents of the cell named D.

66. The instruction PRA 0 (4) prints alphanumeric information on a print line. The contents of the cell whose address is given in B4 is the information which is printed.

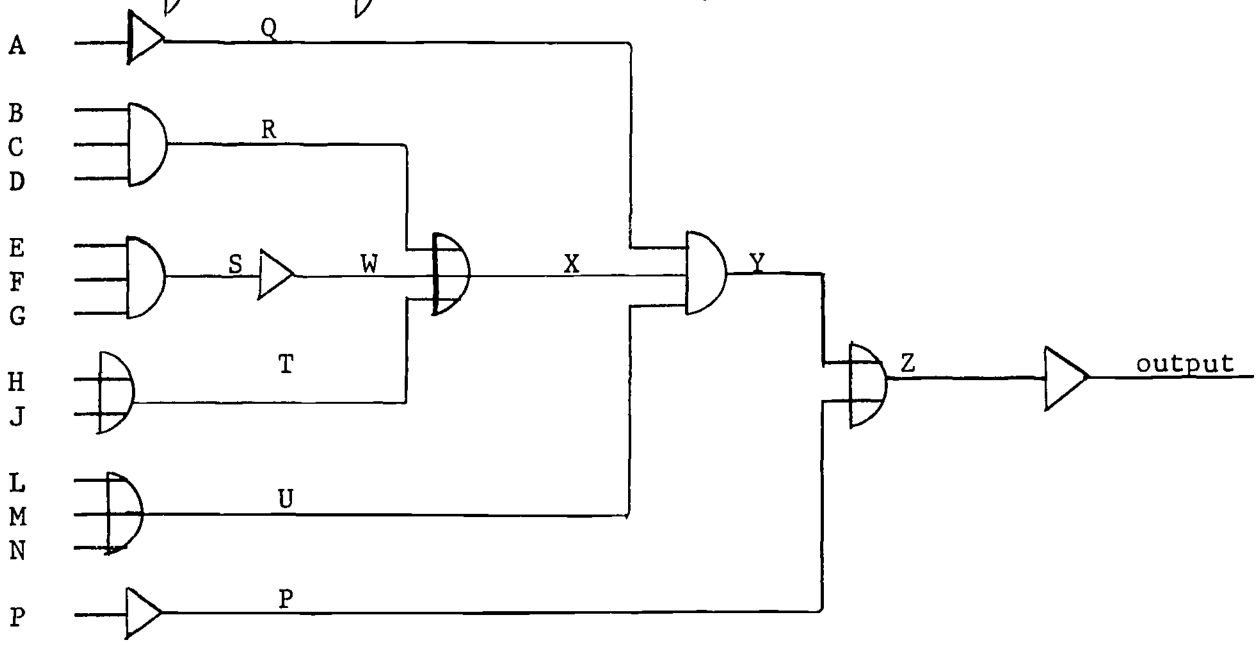
67. All numbers in ELASTIC symbolic instructions are assumed to be decimal unless suffixed by the letter B.

68. Heuristics pertains to a trail and error method for problem solving .

69. A megabit is one thousand bits.

70. A multiprocessor is a computer with more than one arithmetic and logic unit for simultaneous use.

II. Given the following symbols for AND, OR, and INVERTER circuits: AND: OR: and INVERTER: ; and the followong network:



Answer the following questions:

(a). If R = 0, B = 1, and C = 1, then what is the value of D?

71. D = ________

(b). If R = 0, S = 1, and X = 0, then

72. H = ______, 73. J = ______, 74. T = ______, 75. W = ______

(c) If the output of this network is 1 given that X = 0, U = 1, Q = 1, C = 1, D = 1, L = 0, and M = 0, then:

76. A = ______, 77. B = ______, 78. E = ______, 79. F = ______, 80. G = ______

81. H = ______, 82. J = ______, 83. N = ______, 84. P = ______, 85. R = ______

86. S = ______, 87. W = ______, 88. T = ______, 89. V = ______, 90. Y = ______

91. Z = ______.

(d) If W = 0, then: 92. S = ______, 93. E = ______, 94. F = ______, 95. G = ______

Page 6, Final Examination

III. Given the following dummy instructions to work with, answer the following questions true or false.

```
JP1     AMJ JUMP
JP2     AZJ SKIP
HALT    HLT SKIP
RD      RDI DOG
TP1     BSS 1
HTRD    HLT DOG
ADDR    HLT DATA
ONE     DEC 1
CELL    NOP
```

96. To build the instruction UNJ JUMP in the A register, the following instruction sequence can be used.

```
LDA     JP2
ISB     HALT
IAD     JP1
```

97. To build the instruction RDI DATA+2 in the Q register the following sequence of instructions could be used.

```
LDA     RD
ISB     HTRD
IAD     ADDR
IAD     ONE
IAD     ONE
LRS     36
```

98. To build the instruction ISB DATA in the A register, the following sequence of instructions could be used.

```
LDA     RD
ISB     HTRD
ISB     CELL
IAD     ADDR
```

99. To build the instruction LDA DOG in the Q register the following sequence of instructions could be used.

```
LDA     CELL
ISB     JP2
IAD     HALT
IAD     HTRD
```

100. To build the instruction IDV JUMP in the A register, the following sequence of instructions could be used.

```
LDA     JP2
ISB     HALT
IAD     JP1
ISB     CELL
```

Page 7, Final Examination

IV. Answer true or false to each of the following statements.

101. (1A7) sixteen = (423) ten
102. (52)ten = (1221)three
103. (172)five ((62)ten
104. (01100011.1001)two = (3003.21)four
105. 101101 x 101 = 11100011 (binary arithmetic)
106. 101010.11 divided by 11 = 11101.01 (binary arithmetic)
107. 111-1001 = -10 (binary arithmetic)
108. 110110111.1101 + 11111.101 = 111010111.1000 (binary arithmetic)
109. The one's complement of 1101101101 is 0010010010.
110. The two's complement of 101101100 is 1010010100.
111. Quotient overflow occurs when the absolute value of the portion of the dividend contained in the Q register is greater than the absolute value of the divisor.
112. The post-fix form for the expression: A-B*D/E is ABDE/*-.
113. The CDC 6600 computer system has one central processor and 11 peripheral processors.
114. Postfix notation is a notation which facilitiates representation of an expression without the use of parentheses.
115. Expressions which appear in the address field of an ELASTIC instruction are scanned strictly left to right.
116. The character blank in the ELASTIC language is called a delimiter.
117. The size or capacity of an index register in any computer is usually dependent upon the range of addresses in the computer.
118. The capacity of the accumulator register in any fixed-word length computer is usually dependent upon the word size of the computer.
119. In order to simulate a computer with particular characteristics on another computer, a program called an interpreter can be written.
120. When a number is normalized, this usually means that the binary point is placed in a position immediately preceding the most significant 1 of the number.
121. In the ELASTIC computer the coefficient portion of a number in floating-point representation is always greater than or equal to 1/2 and less than or equal to 1.
122. The ELASTIC computer is separated into two parts which are called the assembler and the compiler.
123. All of the instructions in ELASTIC use one address logic.
124. The ELASTIC computer is a binary, sequential machine with a variable word length.
125. The MAC pseudo-op constitutes a portion of a macro heading.

PROGRAM #1

Write a sequence of code in machine language which will read two values from cards into the first two memory locations; that is, 0000 and 0001. Compute the sum of these two numbers and store the sum in location 0002. Print out the two input numbers in decimal, their sum in both octal and decimal, and then halt.

Punch these machine language instructions on cards, one instruction per card. Each instruction should start in column 1 of the card.

Write a program coded in symbolic language which will read these machine language instructions into storage which you have reserved for them and then jump to the first machine language instruction that you read and execute. The two values which you use as input should be the decimal integers 15 and 12.

After reading the machine language instructions, print them out in octal format before jumping to execute them. The last machine language instruction should be the number corresponding to a halt code.

Draw a flow chart for this program and submit it with your output. This program is due the fourth class meeting after it is assigned. Begin working on it as soon as possible so that you will have the maximum time possible for debugging.

Hospitalization is the only acceptable excuse for a late program.

PROGRAM #2

Read five values from cards and store these values in locations named A, B, C, D, and E. Compute the expression $A^2 - 2B + 3C^3 - D/E$. If the value of this expression is greater than 100, print a flag value of 10. If the value of the expression is equal to 100, print a flag value of 15, and if the value of the expression is less than 100, print a flag value of 20. After you have determined whether the expression is greater than, equal to, or less than 100 and printed the appropriate flag value, print the five input values and the value of the computed expression. Perform this procedure for three sets of five cards and then stop. Be sure to extend the sign bit of the dividend before you form the expression D/E. Test your program with the following data sets:

$A = 10,\quad B = 5,\quad C = 2,\quad D = 8,\quad E = 2$

$A = 11,\quad B = 10,\quad C = 1,\quad D = -8,\quad E = -2$

$A = 4,\quad B = 2,\quad C = 1,\quad D = 72,\quad E = 4$

Draw a flow chart for this program and write a documentation according to the form recommended in Chapter VI. of the manual.

This program is due on the fifth class meeting after it is assigned. Begin your work as soon as possible so that you will have the maximum time for debugging. Hospitalization is the only acceptable excuse for a late program.

PROGRAM #3

Perform the same operations as described in Program #2 with the following exceptions:

1. Treat the five values used for the computation of the expression as a linear array instead of using a BSS for each value, that is, your storage reservation should appear as follows:

   NAME BSS 5

2. Write your program with enough generality so that you could handle any number of data sets, five values to a set.
3. Use address modification to input and output values used in the computation.
4. Use the same data as in Program #2 and add three sets of your own choosing.

This program is due on the sixth class meeting after it is assigned. Draw a flow chart for this program and compare the method used with Program #2.

PROGRAM #4

Write Program #3 again using a subroutine to input and output values used in the computation.

PROGRAM #5

Write a program which will satisfy the requirements of the following problem.

A deck of cards is given to you to read. There are two cards for each person in a certain city. The first card contains the person's age and the second card contains his income for 1966. The last card in the data deck contains the number 9999 and should be used to signal the end of the data deck. (This type of procedure is called termination by use of a "trailer card.")

Find the average salary of the people in each five-year group, that is, 0-4, 5-9, 10-14, ..., 95-99. Print out the lower age limit for each group, that is, 0, 5, 10, ..., 95, the average salary for that group and the number of people in each group. Avoid division by zero in case there are no people in an age group.

Write your program with address modification by the use of index registers.

Draw a flow chart for this program and write a documentation according to the documentation outline.

By drawing a good flow chart and developing an efficient algorithm before you begin the actual symbolic coding, you may save a great amount of time debugging. Use remarks to clarify steps in the coding of your program. The program should be general enough to handle any number of cards as input.

Use the following data, one number per card:

6, 100, 50, 20000, 9, 350, 12, 475, 25, 7500, 35, 15000, 65, 4800, 63, 5750, 15, 525, 18, 1800, 16, 1500, 22, 5500, 75, 3000, 72, 3700, 27, 8400, 11, 350, 31, 13000, 37, 18000, 38, 9000, 42, 15000, 47, 63000, 51, 19000, 57, 6500, 26, 10000, 78, 1950, 17, 550, 81, 1250, 84, 150000, 87, 950, 92, 1765, 94, 2575, 99, 5500, 97, 7350, 36, 5750, 45, 27000, 4, 125, 68, 1237, 53, 5300, 9999.

In debugging your program you might try inserting a routine to echo-print the data that you read in to insure that you are working with the correct data.

PROGRAM #7

Write a program which is general enough to perform matrix multiplication for matrices of dimensions up to 10 x 10. You should determine the dimensions of the matrices in any particular case, determine conformability, and compute the product matrix. Print the input matrices and the product matrix as output.

Use a general input/output subroutine to read and print values. Use remarks in your program to clarify and explain the method you are using.

Use any test data that you wish; however, remember that your program should be capable of handling any size matrices up to 10 x 10.

You may use all capabilities of ELASTIC2 and 3 including alphabetic output of labels and headings.

Draw a flowchart for your program that is general enough to enable some other person to write this program in any language he might choose. Write a documentation according to the sheet entitled, "Procedure for Program Documentation."

Be sure that your name and section number are on all work turned in. Begin your work as soon as possible so that you will have the maximum time allowable. Hospitalization is the only excuse for a late program.

**A college algebra text may be consulted for reviewing matrix multiplication.

PROGRAM #10

Write a program which will sort an array of numbers from largest to smallest and from smallest to largest. Your program should be capable of handling any size array up to a maximum of 100 elements. The actual sort routine(s) should be written as macro(s). Read in the array to be sorted using an extended read instruction. Echo-print the array before sorting and then sort the array from largest to smallest and print the array in its sorted form. Sort the array from smallest to largest and print the array in its sorted form. (Do not just print the array backwards.) Print labels above each array before printing.

Draw a flowchart for this program and write a documentation according to the documentation outline. Use test data of your own choosing.

Your flowchart should be general and language independent. You may use all capabilities of ELASTIC to write this program. Begin your work as soon as possible.

PROGRAM #11

The following problem is designed to provide practice in using the ELASTIC3 instruction set and the macro processor.

A message consisting of alphanumeric characters is given to you to be encoded and decoded according to a specified procedure. The message may be as long as 72 characters. You should read the message from a single card and store the message into a work area which you have reserved. Print the message across a line as soon as you have read it.

Each set of six characters should be operated on as a unit. (Notice that this is the maximum amount of information that can be contained in a single ELASTIC word.)

Define a macro which will encode each six-character word according to the following procedure: replace the second character by the sixth, the first by the third, the sixth by the fourth, the fourth by the first, the fifth by the second, and the third by the fifth. This can be denoted in general by representing each character C(i) as follows:

$$\begin{aligned} C(6) &\rightarrow C(2) \\ C(3) &\rightarrow C(1) \\ C(4) &\rightarrow C(6) \\ C(1) &\rightarrow C(4) \\ C(2) &\rightarrow C(5) \\ C(5) &\rightarrow C(3) \end{aligned}$$

Encode the entire message using the above encoding procedure. The macro should be called for the encoding of each word of six characters. When you have encoded the entire message, print it out across the page.

Using the same macro definition, decode the encoded message by changing the calls to the macro. When the entire message has been decoded, print it out across the page.

Use single twelve-word work area to hold the original message that you read, the encoded message, and the decoded message. All printing of the variations of the message should be from this single work area.

Use the following message to test your program:

PACK MY BOX WITH FIVE DOZEN LIQUOR JUGS.

Draw a flow chart for this program and write a documentation according to the documentation outline.

This program is due on the sixth class meeting after it is assigned. Begin your work as soon as possible so that you will have maximum time for debugging. Hospitalization is the only excuse for a late program.

Procedure for Program Documentation

In order to have some formalized guidelines to go by, the following outline is recommended for use in documenting programs. Each program turned in for credit should be organized as follows:

I. Cover Sheet

A. Name and title of program
i.e., PRØGRAM TWØ--a program to sum five numbers

B. Programmer (your name)

C. Date

D. Class and section number

II. Written Documentation

A. Identification of particular program or subprogram

1. Name and title
2. Programmer and date
3. Type of program or subprogram and in which language it is written
4. Original computer and processing system used
5. History of program--list any major versions of the program prior to this revision; if just written, include only the date assigned and the data completed

B. Purpose--a brief statement of what the routine is supposed to do

C. Use of program

1. Operational procedure--what the intelligent programmer needs to know about how to get the program operational. That is, list any special files, tapes, assignments, or control cards that need to be set prior to running the program.
2. Data I/Ø
   a. Input parameters and format--how and where to input which variables.
      Note: In the order in which they are read in, give the variable's name, description of what it is, and where it comes from (i.e., which data card field or tape record).
   b. Output parameters and format--briefly describe <u>how</u> and in what order which variables are printed
3. Cautions to the user--record any special cases where the program's actions are not completely clear.
4. Error provisions--like cautions to the user; but these are errors the program is equipped to find, and which are usually noted by printing out some sort of message. A statement should be included as to whether the error was a fatal one (halt execution) or just a diagnostic-producing error.

D. Miscellaneous information on all subroutines (including macros) and their parameters

1. Description of input parameters--record where and which (by position and dummy argument name) variables input information to the subprogram.

2. Description of output parameters--where the result of the subprograms' work is recorded.
3. Most library function calls should not be recorded, but any "special" (not well known) functions should be.

E. Description of processing procedure--this section is for key information about the program; it can be helpful to think of this section as being a "verbal" flow chart with complete descriptions of the important aspects of your algorithms.

   Example: A program to change from internal to external BCD codes might have a very complex algorithm or use some special array. Here you would include a description of the algorithm and/or the contents of the array plus any background material that was referenced that might aid in modifying the program.

F. Achievement of purpose and efficiency

   1. Testing--here tell briefly what tests were made and what results were achieved from the tests.
   2. Evaluation
      a. Conclusion on method
      b. Analysis of any numerical error that occurred
      c. Analysis and suggestions for improving any routine which is malfunctioning
      d. Discussion of other algorithms that might have been used and/or other ways of coding the current one (this section is required in full).

G. Flow chart--if a subprogram is called only once, its flow chart should be included besides that of the calling program. If one is called many times it probably should be flow-charted separately and included at the end. It is preferable that the student use the ASA standard set of flow-charting symbols.

H. Listing of program--include the listing of all routines and the output

I. Brief summary of changes made, by whom, and data made (added only in case of partial revision after the documentation is complete).

Additional Instructions

Whenever it seems that some aspect or piece of information about your program applies to two or more of the sections mentioned above, it would be wise and highly recommended to include it in all the applicable sections.

A documentation should be neatly handwritten or typed. Be sure all parts are stapled together or bound in some fashion. A manila or braded folder may be used to hold your documentation. The only excuse for a late program is hospitalization; otherwise, no late programs will be accepted.

APPENDIX D

INDEX TO ELASTIC INSTRUCTIONS

ELASTIC1 Instruction Set

Instruction	Octal Code	Description, Function
AMJ y	37	Jump if (A) is negative
AZJ y	36	Jump if (A) = ± 0
HLT y	00	Halt
IAD y	14	Integer add; $(A) + (y) \rightarrow A$
IDV y	25	Integer divide; $[(QA)/(y)]_{quotient} \rightarrow A$ $[(QA)/(y)]_{remainder} \rightarrow Q$
IMU y	24	Integer multiply $(A)\cdot(y) \rightarrow QA$
ISB y	15	Integer subtract; $(A) - (y) \rightarrow A$
LDA y	12	Load A-register; $(y) \rightarrow A$
LDQ y	16	Load Q-register; $(y) \rightarrow Q$
NØP y	50	No operation
PRI y	66	Print (y) as decimal integer
PRØ y	74	Print (y) as octal integer
RDI y	65	Read decimal integer into y
RDØ y	73	Read octal integer into y
STA y	20	Store (A-reg); $(A) \rightarrow y$
STQ y	21	Store (Q-reg); $(Q) \rightarrow y$
UNJ y	75	Unconditional jump to y

****

Pseudo-instructions:

END y	--	End assembly; start execution at location y
BSS k	--	Reserve k storage locations
DEC k	--	Create decimal constant of k

ELASTIC2 Instruction Set

Instruction	Octal Code	Description, Function
ENX y(x)	50	Replace contents of index register x by y. $y \rightarrow (Bx)$
INX y(x)	51	Increase contents of index register x by y. $(Bx) + y \rightarrow (Bx)$
LDX y(x)	52	Replace contents of index register x by the low order 12 bits of the contents of memory location y. $(y_{12}) \rightarrow (Bx)$
STX y(x)	53	Replace the low order 12 bits of the contents of memory location y by the contents of index register x. $(Bx) \rightarrow (y_{12})$
XEJ y(x),b	54	Jump to location y if the contents of index register x equals b.
XHJ y(x),b	55	Jump to location y if the contents of index register x are greater than b.

****

Expressions permitted as addresses. Constants, symbols, +, -, *, and / and index register designations are permitted in the address field.

Absolute address permitted.

ELASTIC3 Instruction Set

Instruction	Octal Code	Description, Function
ARS n(x)	01	Shift (A-reg) right end-off the number of bits specified by the sum of n and the contents of index register x.
QRS n(x)	02	Shift (Q-reg) right end-off the number of bits specified by the sum of n and the contents of index register x.
LRS n(x)	03	Shift (AQ-reg) right end-off the number of bits specified by the sum of n and the contents of index register x.
ALS n(x)	05	Shift (A-reg) left end-around the number of bits specified by the sum of n and the contents of index register x.
QLS n(x)	06	Shift (Q-reg) left end-around the number of bits specified by the sum of n and the contents of index register x.
LLS n(x)	07	Shift (AQ-reg) left end-around the number of bits specified by the sum of n and the contents of index register x.
ENA n(x)	10	Enter the sum of n and the contents of index register x into the low order 12 bits of the A-register. Upper 24 bits of the A-register are copies of the high order bit of the original 12-bit quantity. $n + (Bx) \rightarrow (A\text{-register})$
INA n(x)	11	Increase (A-reg) by the sum of n and the contents of index register x. $(A\text{-register}) + n + (Bx) \rightarrow (A\text{-register})$
COM	40	Complement the contents of the A-register $(A\text{-register}) \rightarrow (A\text{-register})$
ANA y(x)	41	Form the logical product of the contents of the A-register and the contents of the word whose address is given by the sum of y and the contents of index register x. $(A\text{-register}) \cdot (y+(Bx)) \rightarrow (A\text{-register})$

Instruction	Octal Code	Description, Function
ORA y(x)	42	Form the logical sum of the contents of the A-register and the contents of the word whose address is given by the sum of y and the contents of index register x. (A-register) + (y+(Bx)) → (A-register)
RDA y(x)	71	Read six columns from a card into display code and place this display code in the memory location whose address is y + (Bx).
PRA y(x)	72	Print (y+(Bx)) on a line as 6 display coded characters

****

Extended Input/Output

RDI y(x),z(w)	65	Read decimal values from a card. The first value is read into y + (Bx). Continue reading values until a value is read into z + (Bw). Values separated on the card by commas.
RDO y(x),z(w)	73	Read octal values from a card. The first value is read into y + (Bx). Continue reading values until a value is read into z + (Bw).
RDA y(x),z(w)	71	Read Hollerith characters from a card, 6 characters at a time. The first 6 characters are read into y + (Bx). Continue reading until 6 characters are read into z + (Bw).
PRI y(x),z(w)	66	Print decimal values, 7 values per line. The first value is obtained from y + (Bx). Continue printing until the value in z + (Bw) is printed on a line.
PRO y(x),z(w)	74	Print octal values, 7 values per line. The first value is obtained from y + (Bx). Continue printing until the value in z + (Bw) is printed on a line.
PRA y(x),z(w)	72	Print Hollerith characters on a line, 108 characters per line. The first six characters are obtained from y + (Bx). Continue printing until the six characters at z + (Bw) have been printed.

****

Pseudo-instructions

Instruction	Description, Function
CON k1,k2,...,kn	Create decimal and octal constants. Octal constants are suffixed by "B". Multiple constants created by a single CON should be separated by commas.
CON Hcccccc,cccccc,...,cccccc	Create Hollerith constants. Constants are treated as 6-character groups separated by commas

****

Current instruction address, denoted "$", is available for use in addresses. Expressions are permitted in the address field of BSS, DEC, and END.

Macro Processor Instructions

name MAC P1,P2,...,P9	Create a macro definition using Pi (i=1,9) as formal parameters in the prototype.
name END	Terminate the macro being defined.
name MOP (arg1,arg2,...,arg9)	Call the macro whose name appears in the location field. Substitute actual parameters arg1,arg2,...,arg9 in the macro prototype.
IFT m,p,n	If m = p, assemble the following n lines of code; otherwise, skip n lines of code.
IFF m,p,n	If m ≠ p, assemble the following n lines of code; otherwise, skip n lines of code.

****

Macro facilities available in all ELASTIC computers.

APPENDIX E

OVERVIEW OF PICLS INSTRUCTIONAL SYSTEM

PICLS (Purdue Instructional and Computational Learning System) is a CAI instructional system which was developed at Purdue University [25]. A variation of PICLS was implemented on the CDC 6600 computer system at The University of Texas at Austin during the Fall of 1969.

The PICLS system is primarily oriented toward using a teletype in conversational mode; however, a batch mode is also available. In the batch mode, the commands which are ordinarily issued at the teletype, are input on punched cards. An entire session at the teletype may be simulated in batch mode in this way. The character set for PICLS is the same as for a standard Model 33 teletypewriter.

As it exists on the CDC 6600, the PICLS system is executed in conversational mode by utilizing the CONVERSE feature of the RESPOND time-sharing system. It is for this reason then that the teletype driver for transmission of text and responses is embedded in RESPOND instead of PICLS. When a person wishes to use PICLS, he must first log onto the RESPOND system and then enter PICLS through the CONVERSE feature. Once in the conversational mode, the user must log onto the PICLS system according to the course with which he is interested. It should be mentioned in passing that an instructional course written in the PICLS author language is divided into a number of sections. Each section is referenced by a unique name assigned to it by the author.

The PICLS system consists of a set of file management routines for creation, deletion, and manipulation of files as well as a compiler and interpreter for processing instructional programs written in the PICLS author language. There are three types of users relative to the file management routines: system users, authors, and students. Each type of user has access to a different set of the file management commands. A student may issue student commands, an author may issue student and author commands, and a system user may issue any PICLS command.

The syntax for elements of PICLS commands is presented below in Backus Normal Form.

SYNTAX FOR ELEMENTS OF PICLS COMMANDS

⟨letter⟩ ::= A|B|C ... |Z

⟨digit⟩ ::= 0|1|2|3| ... |9

⟨letter-or-digit⟩ ::= ⟨letter⟩|⟨digit⟩

⟨number⟩ ::= ⟨digit⟩|⟨number⟩⟨digit⟩ (6 or fewer digits)

⟨section-name⟩ ::= ⟨letter⟩|⟨section-name⟩⟨letter-or-digit⟩ (6 or fewer characters, 1st has to be alphabetic)

⟨file-name⟩ ::= ⟨letter⟩|⟨file-name⟩⟨letter-or-digit⟩ (7 or fewer characters, 1st alpha)

⟨yes-or-no⟩ ::= YES|NO|Y|N

⟨n1⟩ ::= ⟨number⟩

⟨n2⟩ ::= ⟨number⟩

⟨parameter-list⟩ ::= ⟨parameter⟩|⟨parameter-list⟩⟨parameter⟩

⟨command⟩ ::= $⟨command-word⟩|$⟨command-word⟩,⟨parameter-list⟩

⟨command-word⟩ ::= LOGOFF|ENDFUN|LESSON|INPUT|EDIT|FILE|COMPILE|PRINT|DELETE RENUMBER|LIST|FDUMP|SCRATCH|FLIST|ABOUT|SECDUMP

⟨parameter⟩ ::= ⟨number⟩|⟨section-name⟩|⟨file-name⟩|⟨yes-or-no⟩⟨n1⟩|⟨n2⟩|ALL NONE

The PICLS commands will be briefly described below according to user type.

Student Commands

$LOGOFF - logs the current user off and recycles to the LOGON procedure to log a new user onto the system.

$ENDRUN - logs the current user off and terminates execution of PICLS.

$LESSON,⟨section-name⟩ - initiates execution of an instructional program at a particular section.

Author Commands

The author commands include the student commands listed above and the commands described below.

$INPUT,⟨section-name⟩,⟨n1⟩,⟨n2⟩ - used to enter author language statements into a section. ⟨section-name⟩ denotes the section of interest while ⟨n1⟩ is the number to be given to the first record and ⟨n2⟩ is the line increment number.

$EDIT,⟨section-name⟩ - used to modify statements in a section. Replacement, insertion and deletion operations are available in EDIT mode.

$FILE - causes all modifications to the section being built to be recorded permanently.

$COMPILE,⟨section-name⟩,⟨yes-or-no⟩,⟨yes-or-no⟩ - causes the PICLS compiler to compile the denoted section and prepare it for execution. The first ⟨yes-or-no⟩ requests or rejects production of a compiled section listing while the second ⟨yes-or-no⟩ specifies whether or not to list the symbol table for the section. A section must be compiled without errors before it can be executed with a $LESSON command.

$PRINT,⟨section-name⟩, ALL - prints all records of a section.

$PRINT,⟨section-name⟩,⟨n1⟩,⟨n2⟩ - prints from record ⟨n1⟩ to record ⟨n2⟩ of a section.

$PRINT,⟨section-name⟩, NONE - prints the status (compiled or not) of a section.

$DELETE,⟨section-name⟩ - deletes denoted section from user's files.

$RENUMBER,⟨section-name⟩ - renumbers the lines of a section with 10 and using increments of 10.

$LIST - causes a listing of all sections in the user's files.

$FDUMP - causes working versions of a user's files to be copied to permanent storage.

System Commands

The system commands include all student and author commands as well as those listed below.

$SCRATCH,⟨file-name⟩ - deletes denoted file from the PICLS system.

$FLIST - produces list of all files in PICLS system.

$ABORT - aborts current PICLS run.

$SECDUMP,⟨section-name⟩,⟨yes-or-no⟩, ALL - produces a structured dump on the teletype of the denoted section. If YES is given the symbol table is also produced. ALL causes a dump of the entire section.

$SECDUMP,⟨section-name⟩,⟨yes-or-no⟩,⟨n1⟩,⟨n2⟩ - same as above except that records ⟨n1⟩ to ⟨n2⟩ of the denoted section are dumped.

The PICLS author language is used to write instructional courses. The language is very similar to IBM's COURSEWRITER I language; however, it differs markedly in the areas of logical decision-making, branching, and facilities for computation. Development of an instructional course is accomplished by the author's creation of a number of sections. Each section is composed of up to 350 statements, and these sections are then connected to form a course. Statements in a section may be uniquely labeled by the author for future reference.

The syntax for author language statements as represented in Backus Normal Form is as follows:

⟨statement⟩ ::= ⟨label⟩:⟨op-code⟩:⟨variable-field⟩:⟨transfer⟩

The ⟨label⟩ and ⟨transfer⟩ are optional; however, ⟨op-code⟩ is always required and the presence of ⟨variable-field⟩ depends on the particular ⟨op-code⟩. The execution of statements may be explicitly determined by the author's use of ⟨transfer⟩ fields or implicitly controlled by the success or failure of executing a certain op-code as in COURSEWRITER I. Implicit sequencing may be determined by the fact that all "major" operations are executed while the execution of a sequence of "minor" operations is skipped if the preceding "major" operation produced a failure condition.

The formal syntax for the PICLS author language is presented in Backus Normal Form below.

FORMAL SYNTAX OF THE AUTHOR LANGUAGE

⟨letter⟩ ::= A | B | C | ... | Z
⟨digit⟩ ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
⟨name⟩ ::= ⟨letter⟩ | ⟨name⟩ ⟨letter⟩
    Note: a name consists of six or fewer characters
⟨integer⟩ ::= ⟨digit⟩ | ⟨integer⟩ ⟨digit⟩
⟨fraction⟩ ::= . ⟨integer⟩

⟨unsigned-number⟩ ::= ⟨integer⟩ | ⟨fraction⟩ | ⟨integer⟩ ⟨fraction⟩
Note: an unsigned number must consist of 35 or fewer characters, including the decimal point, if any

⟨number⟩ ::= ⟨unsigned-number⟩ | +⟨unsigned-number⟩
| -⟨unsigned-number⟩

⟨simple-variable⟩ ::= ⟨name⟩

⟨subscript-list⟩ ::= ⟨arithmetic-expression⟩ | ⟨subscript-list⟩
⟨arithmetic-expression⟩

⟨subscripted-variable⟩ ::= ⟨name⟩[⟨subscript-list⟩]

⟨variable⟩ ::= ⟨simple-variable | ⟨subscripted-variable⟩

⟨argument-list⟩ ::= ⟨arithmetic-expression⟩ | ⟨argument-list⟩
⟨arithmetic-expression⟩

⟨arithmetic-function⟩ ::= ⟨name⟩ (⟨argument-list⟩)

⟨primary⟩ ::= ⟨number⟩ | ⟨simple-variable⟩ | ⟨subscripted-variable⟩
| ⟨arithmetic-function⟩ | (⟨arithmetic-expression⟩)

⟨factor⟩ ::= ⟨primary⟩ | ⟨factor⟩ ↑ ⟨primary⟩

⟨multiplying-operator⟩ ::= * | /

⟨term⟩ ::= ⟨factor⟩ | ⟨term⟩ ⟨multiplying-operator⟩ ⟨factor⟩

⟨adding-operator⟩ ::= + | -

⟨arithmetic-expression⟩ ::= ⟨term⟩ | ⟨adding-operator⟩
⟨arithmetic-expression⟩ | ⟨arithmetic-expression⟩
⟨adding-operator⟩ ⟨term⟩

⟨relational-operator⟩ ::= 'LT' | 'LE' | 'EQ' | 'NE' | 'GE' | 'GT'

⟨relation⟩ ::= ⟨aritimetic-expression⟩ ⟨relational-operator⟩
⟨arithmetic-expression⟩

⟨logical-primary⟩ ::= ⟨relation⟩ | (⟨logical-expression⟩)

⟨logical-factor⟩ ::= ⟨logical-primary⟩ | 'NOT' (⟨logical-expression⟩)

⟨logical-term⟩ ::= ⟨logical-factor⟩ | ⟨logical-term⟩ 'AND'
⟨logical-term⟩

⟨text⟩ ::= ⟨any character but :⟩

⟨system-function⟩ ::= ⟨name⟩ (⟨text⟩)

⟨bound-list⟩ ::= ⟨integer⟩ | ⟨bound-list⟩ , ⟨integer⟩
Note: an integer in a bound list must be greater than zero

⟨array-declaration⟩ ::= ⟨name⟩ [⟨bound-list⟩]

⟨external-reference⟩ ::= ⟨name⟩ | ⟨name⟩ , ⟨name⟩

⟨label⟩ ::= ⟨name⟩

⟨op-code⟩ ::= AA | AN | AR | BR | CA | CB | CN | CR | ES | EX | FJ
| FN | IF | NO | QU | RD | ST | TY | UN | WA | WB | WN | WR

⟨arithmetic-assignment⟩ ::= ⟨variable⟩ = ⟨arithmetic-expression⟩

⟨variable-field⟩ ::= ⟨text⟩ | ⟨arithmetic-expression⟩ |
⟨arithmetic-assignment⟩ | ⟨logical-expression⟩ |
⟨system-function⟩ | ⟨array-declaration⟩ | ⟨external-reference⟩

⟨unconditional-transfer⟩ ::= (⟨label⟩)

⟨conditional-transfer⟩ ::= S (⟨label⟩) | F (⟨label⟩) | S (⟨label⟩)
F (⟨label⟩) | F (⟨label⟩) S (⟨label⟩)

⟨transfer⟩ ::= ⟨unconditional-transfer⟩ | ⟨conditional-transfer⟩

⟨basic-statement⟩ ::= : ⟨op-code⟩ | : ⟨op-code⟩ : ⟨variable-field⟩

⟨unlabeled-statement⟩ ::= ⟨basic-statement⟩ | ⟨basic-statement⟩ :
⟨transfer⟩

⟨statement⟩ ::= ⟨unlabeled-statement⟩ | ⟨label⟩ ⟨unlabeled-statement⟩

⟨comment⟩ ::= * ⟨any-character-string⟩

A discussion of the interpretation of various parts of an author language statement is presented below.

⟨label⟩ - represents the name for a statement. Reference to a statement label may be made in the ⟨transfer⟩ field of other statements in a section or in the variable field of a BR statement.

⟨op-code⟩ - specifies a particular operation to be performed by the computer.

⟨variable-field⟩ - information whose use is determined by the particular ⟨op-code⟩.

⟨unconditional-transfer⟩ - specifies to transfer control unconditionally to another statement in the section if execution of the ⟨op-code⟩ resulted in a "success" condition. The ⟨transfer⟩ field will appear as :(label) .

⟨conditional-transfer⟩ - specifies a branch on "success' and/or a branch on "failure." The ⟨transfer⟩ field will appear as: S(label) or: F(label) or: S (label)F(label) .

The major operations in the PICLS author language are described below.

<u>Major Operations</u>

:QU:⟨text⟩ - presents text to student and waits for a response from the student which when given causes control to proceed to the next statement in sequence.

:RD:⟨text⟩ - presents text to the student and waits for a student response which when given causes control to proceed to the next sequential statement.

:CA:⟨text⟩
:CA:⟨text⟩:⟨transfer⟩ - specifies a correct answer to a question asked in a QU. If the ⟨text⟩ matches the student response a "success" condition

exists, otherwise a "failure" condition is present. If a success condition exists, all following minor operations are executed unless a "success" transfer is specified. If no success transfer is given, control will proceed to the next QU or RD. If a "failure" condition exists, an unconditional or failure transfer is taken if present; otherwise, all following minor operations are skipped and control proceeds to the next major operation.

:CB:⟨text⟩ - specifies an alternate correct answer. If matched, the same operations as specified for the preceding CA will be performed.

:CN:⟨arithmetic-expression⟩ - compares student response with the specified
:CN:⟨arithmetic-expression⟩:⟨transfer⟩
arithmetic expression. Processing on success or failure conditions is the same as explained for the CA operation.

:CR:⟨logical-expression⟩
:CR:⟨logical-expression⟩:⟨transfer⟩ - compares the student response with the ⟨variable field⟩. Processing for success or failure is the same as for the CA operation.

:WA:⟨text⟩
:WA:⟨text⟩:⟨transfer⟩ - specifies an anticipated wrong answer to the previous question. If a match between the text and the student response occurs, a success condition exists and processing proceeds in an analogous manner to the success condition for a CA except that control is passed back to a point immediately following the preceding QU and another response is required. The failure condition receives analogous processing as that for failure of a CA.

:WB:⟨text- - specifies an alternate wrong answer. A match causes execution of the WA sequence directly above the WB.

:WN:⟨arithmetic-expression⟩ - specifies a wrong number for an answer. Processing is analogous to that for a WB.

:WR:⟨logical-expression⟩ - specifies a wrong logical expression. Processing is analogous to that for WB.

:AA:⟨text⟩
:AA:⟨text⟩:⟨transfer⟩ - specifies an anticipated answer that is neither right or wrong. Processing is the same as for the WA operation.

:AN:⟨arithmetic-expression⟩
:AN:⟨arithmetic-expression⟩:⟨transfer⟩ - specifies an anticipated arithmetic expression that is neither right or wrong. Processing is the same as for WA.

⟨UN:⟨text⟩
⟨UN:⟨text⟩:⟨unconditional-transfer⟩ - specifies action when the student response is unrecognizable. Each occurrence of an unrecognizable answer is processed by a different UN as each UN is executed only once. When a UN is executed, the ⟨text⟩ is presented to the student and all minor operations immediately following the UN are executed unless an ⟨unconditional-transfer⟩ appears in which case a transfer is initiated. If no transfer exists, control is returned to a point immediately after the preceding QU and a new response is awaited. If all UNs for a particular QU are executed and another unrecognizable response is received, control proceeds to the next sequential major operation.

:NO
:NO⟨unconditional-transfer⟩ - serves as a place-holder which causes a transfer of execution of all following minor operations.

:XO: - much like NO but takes the place of an RD or QU when more than 10 lines are types at one time before requesting a student response.

:FJ:⟨name⟩
:FJ:⟨name⟩:⟨transfer⟩
:FJ:⟨name⟩(⟨text⟩)
:FJ:⟨name⟩(⟨text⟩):⟨transfer⟩
- requests the performance of a PICLS system function. The text field is used in conjunction with the function. If the function succeeds, all following minor operations are executed or a success transfer is made if specified. If the function fails, all minor operations are skipped and control passes to the next major operation unless a failure transfer is specified in which case it is taken.

:IF:⟨logical-expression⟩
:IF:⟨logical-expression⟩:⟨transfer⟩ - used for logical decision-making. If the logical expression when evaluated is true, all minor operations after the IF are executed and control proceeds to the next major operation unless a success or unconditional-transfer is given, in which case control proceeds to the designated statement. If the logical expression evaluates as false, a failure or unconditional transfer is taken; otherwise all following minor operations are skipped and control proceeds to the next major operation.

:AR:⟨name⟩[⟨bound-list⟩] - declares an array to be associated with a variable name. The ⟨bound list⟩ declares maximum values for subscripts.

:EX:⟨section-name⟩
:EX:⟨section-name⟩⟨label⟩ - transfers control to first statement of another section named as the ⟨section-name⟩ parameter and if ⟨label⟩ is given also, the statement having that label in the named section is given control. Control does not return to the section where the external reference occurred.

:ES
:ES:⟨section-name⟩
:ES:⟨section-name⟩,⟨label⟩
- declares the end of a section. The next section to be executed is denoted by the contents of the ⟨variable-field⟩.

Minor Operations

:BR:⟨unconditional-transfer⟩ - causes control to transfer to the statement whose label appears in the ⟨unconditional-transfer⟩ field.

:TY:⟨text⟩
:TY:⟨text⟩:⟨unconditional-transfer⟩ - causes the ⟨text⟩ to be output to the terminal and a transfer taken if specified.

:FN:⟨name⟩
:FN:⟨name⟩:⟨transfer⟩
:FN:⟨name⟩(⟨text⟩)
:FN:⟨name⟩(⟨text⟩):⟨transfer⟩
- like FJ but acts like a minor operation

:ST:⟨arithmetic-assignment⟩
:ST:⟨arithmetic-assignment⟩:⟨unconditional-transfer⟩ - sets the value of a simple or subscripted variable to the value specified on the right of the equal sign. A transfer is taken if denoted.

APPENDIX F

OVERVIEW OF THE RESPOND TIME-SHARING SYSTEM

The RESPOND system is used to facilitate time-sharing on the CDC 6600 computer at The University of Texas at Austin [24]. The current time-sharing implementation provides remote access to the CDC 6600 through the use of telecommunications equipment and remote terminal teletypewriters. Normally the remote terminal will be a Model 33 or Model 35 teletypewriter which is connected to the CDC 6600 through an acoustic coupler or 101c Data Set and voice-grade telephone line.

The remote user utilizes the features of RESPOND to create, manipulate, destroy, and display files as well as to submit programs to the CDC 6600 for execution. There are three modes of operation available on RESPOND: a command mode for issuing specific directions to the RESPOND system; a data mode for creating records which will be collected into files; and a message mode for communicating with personnel at the central computer site. Through RESPOND the user may create permanent copies of files on punch cards, microfilm, paper tape, magnetic tape, or printer listings. It is significant to note that any computer operation or function performed in batch (over-the-counter) mode can also be accomplished using RESPOND in a time-sharing environment.

Since the purpose of this appendix is to present an overview of the RESPOND system, only the more frequently used commands will be described below. There are many variations of the different commands, so rather than attempt to enumerate every variation, each command will be listed and discussed relative to its general function. Each command will be denoted by listing its name in capital letters.

In order to clarify the explanation of some commands it is necessary to briefly explain the different types of files and formats available to a user. Within RESPOND there are a number of public files which are available to all users. Also accessible to all users are a collection of public formats which are used to create records or data according to a specified form. In addition to public files and formats each user can create private files and private formats which are associated with his particular password. Only users of the password can access these private files and formats.

Communication Commands

LOGIN is used to gain entrance to the RESPOND system. After typing LOGIN the user gives his password and account number for assigning computer charges.

LOGOUT is used to terminate a session using RESPOND. The LOGOUT command causes all files created during a session to be permanently saved.

MSG is used to communicate with personnel at the central computer site.

File Maintenance Commands

CLEAR is used to erase the user's working storage in computer memory. This command should be given prior to building or modifying a file.

DELETE is used to destroy a private file, a private format or portions of a private file. If only records are deleted, the file must be loaded into the user's job table (working space in computer storage).

DISPLAY is used to cause a listing of all or parts of a public or private file. A file must be loaded into the user's job table before it can be displayed.

ENTER is used to put RESPOND in the data mode. It is in the data mode that records and files can be created and modified. Depending upon the form of the ENTER command, the records to be entered by the user can be automatically or manually sequenced. Records can be formatted according to a public or private format or entered in a nonformatted mode.

FILE is used to create a file consisting of all or parts of the current contents of the user's job table. Through the use of this command, files can be merged or combined in any fashion.

FORMAT is the command which is used to create a user-defined private format. Using a format command, the record length, first record and record increment, tab positions and skip positions can be specified.

LIST is used to obtain a listing of the public file/format catalogue and/or the user's private file/format catalogue. Included in this listing is the file or format name, the number of records occupied by the file, the record size, and the date of creation of the file or format.

FLST is a form of the LIST command which causes a shortened listing for each file in which only the file name and number of records occupied by the file are given.

LOAD is used to cause the system-required information associated with a file to be entered in the user's job table. After a file has been loaded, it may be displayed, modified, or manipulated.

SAVE causes an automatic LOGOUT and LOGIN so that all work done by the user during a particular session will be recorded permanently.

SHOW performs the function of LOAD and DISPLAY using a single directive. The main difference between issuing DISPLAY and SHOW is that the record numbers assigned to records in a file may be listed with a DISPLAY command while no such facility exists for the SHOW command.

Break Command

"CTRL""BELL" The break command is used to interrupt or terminate certain activities. The context in which the command is given determines the action taken. The command is issued by striking the "CTRL" key and "BELL" key simultaneously. An example of using the break command occurs when the user desires to switch from the data mode to the command mode.

Batch Processing Commands

COMPILE/ASSEMBLE is used to cause compilation or assembly of a user file. The binary object code that results from compilation or assembly may be retrieved and later executed by the user.

COPY is used to create a punched card deck or printed listing of a file. Other forms of the COPY command allow transfer of files to and from a "COMMON" area so that better utilization of a user's disk space may be realized.

SUBMIT is used to affect execution and/or compilation of a user's file(s). Both binary and coded files may be specified in a SUBMIT command. The user may designate data files, program files, and whether or not the output file(s) are to be recovered as private files.

STATUS is normally used to determine the status of a previously "SUBMITTED" job. The RESPOND system will reply with a designation that the job is in the RESPOND queue, the CDC 6600 system input queue, in execution, in the CDC 6600 system output queue, or that the job is not in the system (has never been submitted or has been completely executed).

Conversation Mode

CONVERSE is a special form of the SUBMIT command which enables a user to communicate with a running job in "conversational mode." It is through the use of the CONVERSE feature that the PICLS instructional system operates.

A sample session using RESPOND appears in Appendix B and in Appendix I. The paper to reference in Appendix B is entitled "ELASTIC on RESPOND." By examining this paper the use of many of the commands explained above may be clarified.

APPENDIX G

EXCERPTS FROM STUDENT TYPEOUTS

On the following pages appear excerpts from the CAI course as it was presented to each student during the summative evaluation. The underlined portions denote student entered information while all other text is output either by PICLS, the CAI program, or RESPOND.

```
  * PICLS READY FOR LOGON
 TYPE USER NUMBER
* 3
 TYPE PASS WORD
*FHOM
 LOGON FOR COURSE ELASTO3 AT 14.48.31 HOURS ON 19 JAN 7
 WAIT FOR LOAD OF INDEXED FILES
    3 FILES LOADED WITHOUT ERROR
 END OF INDEXED FILE LOAD
 TYPE COMMAND
*$LESSON,BEGIN
BEFORE BEGINNING TO LEARN THE ELASTIC LANGUAGE IT MIGHT BE
HELPFUL TO KNOW SOME OF THE CHARACTERISTICS OF THE ELASTIC
COMPUTER.  THE ELASTIC COMPUTER IS THE MACHINE WHICH
EXECUTES YOUR PROGRAMS WRITTEN IN THE ELASTIC LANGUAGE.
PRESS RETURN WHEN YOU HAVE READ THIS.
*
THE ELASTIC COMPUTER IS A BINARY, SEQUENTIAL MACHINE.  IT
HAS 4096 WORDS OF MEMORY WITH AN ADDRESS RANGE IN OCTAL,
FROM 0000 TO 7777.  THE WORD LENGTH IS 36 BITS.  THE
ELASTIC COMPUTER HAS BINARY, FIXED-POINT ARITHMETIC
PERFORMED MODULO 2**36-1, THAT IS, ONE'S COMPLEMENT
ARITHMETIC.  IT HAS CARD INPUT AND LINE PRINTER OUTPUT.  THE
 ELASTIC COMPUTER HAS AN A-REGISTER AND A Q-REGISTER, EACH
36 BITS IN LENGTH.  HOW MANY WORDS OF MEMORY DOES THE
ELASTIC COMPUTER POSSESS?
*4096 WORDS
THAT'S CORRECT.
HOW LONG IS A WORD IN ELASTIC?
*36 BITS
THAT'S RIGHT, 36 BITS LONG.
THE WORD BIT IS A CONTRACTION OF THE WORDS BINARY DIGIT.
SINCE THERE ARE 36 BITS IN AN ELASTIC COMPUTER WORD, WHAT
IS THE LARGEST OCTAL NUMBER WHICH CAN BE STORED IN A WORD?
(REMEMBER THAT IT TAKES 3 BITS TO REPRESENT AN OCTAL
DIGIT.)
*777777777777
VERY GOOD.
WHAT IS THE INPUT MEDIUM FOR THE ELASTIC COMPUTER, THAT IS,
CARDS, TAPE, PRINTED LISTING, ETC.?
*PUNCHED CARDS
THAT'S CORRECT.
WHAT TYPE OF OUTPUT IS PRODUCED BY ELASTIC?
*PRINTED LISI←TING
VERY GOOD.
WHEN WE SAID THAT ARITHMETIC IN ELASTIC WAS ONE'S
COMPLEMENT, WE MEANT THAT NEGATIVE NUMBERS ARE REPRESENTED
INTERNALLY AS THEIR COMPLEMENT.  IN ELASTIC, WHAT IS THE
OCTAL REPRESENTATION FOR THE NUMBER -1246?
*777777776531
THAT'S CORRECT.
A STATEMENT IN ELASTIC IS COMPOSED OF 4 FIELDS; LOCATION
FIELD, OPERATION CODE FIELD, ADDRESS FIELD AND REMARKS
FIELD.  DEPENDING ON THE PARTICULAR STATEMENT, ONE OR MORE
OF THESE FIELDS MAY BE BLANK.  PRESS RETURN.
```

*

COLUMNS 1 THROUGH 6 CONSTITUTE THE LOCATION FIELD. THE LOCATION FIELD MUST BE EITHER BLANK OR CONTAIN A LEFT-JUSTIFIED ELASTIC SYMBOL. A GIVEN SYMBOL MUST OCCUR IN THE LOCATION FIELD OF ONE AND ONLY ONE STATEMENT IN THE PROGRAM. WHAT FIELD ARE WE DISCUSSING?

*LOCATION FIELD

IS IT LEGAL FOR A SYMBOL TO BEGIN IN COLUMN 3 OF THE LOCATION FIELD? ANSWER YES OR NO.

*NO

THAT'S RIGHT, THE SYMBOL MUST BE LEFT-JUSTIFIED.
CAN A LEGAL ELASTIC CONSTANT APPEAR IN THE LOCATION FIELD?

*NO

VERY GOOD, ONLY SYMBOLS CAN APPEAR IN THE LOCATION FIELD. COLUMNS 8 THROUGH 10 CONSTITUTE THE OPERATION CODE FIELD. THIS FIELD MUST BE EITHER BLANK OR CONTAIN THE OP OF AN INSTRUCTION. CAN THE OPERATION CODE FIELD EVER BE BLANK?

*NO

YES, THE OPERATION CODE FIELD CAN BE BLANK. TYPE YES.

*YES

THAT'S RIGHT THE OPERATION CODE FIELD CAN BE BLANK. COLUMNS 12 THROUGH 28 CONSTITUTE THE ADDRESS FIELD. THIS FIELD MUST BE EITHER BLANK OR CONTAIN, LEFT-JUSTIFIED, AN R CORRESPONDING TO THE OP IN THE OPERATION CODE FIELD. IF R IS AN ELASTIC SYMBOL THAT SYMBOL MUST OCCUR IN THE LOCATION FIELD OF SOME STATEMENT IN THE PROGRAM. WHAT FIELD ARE WE DISCUSSING?

*OP CODE

NO , IT IS NOT OP CODE. TRY AGAIN.

*ADDRESS

THAT'S CORRECT.
CAN A CONSTANT APPEAR IN THE ADDRESS FIELD? REPLY YES OR NO

*YES

THAT'S RIGHT. IF THE OP CODE REQUIRES THAT A CONSTANT APPEAR IN THE ADDRESS FIELD.
IF A SYMBOL APPEARS IN THE ADDRESS FIELD, IT MUST BE LEFT-JUSTIFIED. MUST THE SYMBOL ALSO APPEAR IN SOME LOCATION FIELD IN THE PROGRAM?

*YES

THAT'S CORRECT. SYMBOLS ARE USED TO IDENTIFY PARTICULAR INSTRUCTIONS OR LOCATIONS AND HENCE WHEN A SYMBOL APPEARS IN THE ADDRESS FIELD IT REFERS TO SOME STORAGE LOCATION THAT HAS BEEN INDICATED OR IDENTIFIED.
COLUMNS 30 THROUGH 80 CONSTITUTE THE REMARKS FIELD. THE CONTENTS OF THIS FIELD ARE COMPLETELY IGNORED BY THE ASSEMBLER HENCE THE FIELD MAY CONTAIN ANY STANDARD KEYPUNCH CHARACTERS. PRESS RETURN

*

COLUMNS 7, 11, AND 29 MUST CONTAIN BLANKS. THESE COLUMNS SERVE AS FIELD SEPARATORS. GIVE THE NUMBER OF THE COLUMN THAT SEPARATES THE LOCATION FIELD FROM THE OP CODE FIELD

*7

VERY GOOD.

WHAT COLUMN SEPARATES THE OP CODE FIELD AND THE ADDRESS
FIELD?
*11
THAT'S RIGHT.
WOULD THE CHARACTERS =,≠,., , , BE LEGAL IN THE REMARKS
FIELD? ANSWER YES OR NO.
*YES
THAT'S CORRECT. SINCE COLUMNS 30 THROUGH 80 ARE IGNORED BY
THE TRANSLATION SYSTEM, ANY KEYPUNCH CHARACTER CAN APPEAR
IN THE REMARKS FIELD.
A DIAGRAM OF THE ELASTIC SYMBOLIC INSTRUCTION FORMAT
APPEARS AS FOLLOWS;

```
                  11 1      22 3         8
1      67 8       01 2      89 0         0
***************************************************
*LOCATION*  *OP CODE*  *ADDRESS*  *REMARKS FIELD*
*  FIELD *  * FIELD *  * FIELD *  *              *
***************************************************
```

PRESS RETURN WHEN YOU HAVE STUDIED THIS DIAGRAM.
*
DO YOU WISH TO GO ON TO THE NEXT
SECTION (TYPE ≠YES≠ OR ≠NO≠)
*YES
THE ELASTIC COMPUTER PERFORMS ARITHMETIC WITH THE HELP OF
TWO REGISTERS. THESE REGISTERS ARE CALLED THE ACCUMULATOR
REGISTER AND THE MULTIPLIER-QUOTIENT REGISTER. THE
ACCUMULATOR IS COMMONLY CALLED THE A-REGISTER WHILE THE
MULTIPLIER-QUOTIENT REGISTER IS CALLED THE Q-REGISTER.
THESE REGISTERS SERVE THE ELASTIC COMPUTER BY ACTING AS AN
≡ADDING MACHINE≡ WHERE INTERMEDIATE RESULTS MAY BE
COMPUTED. WHAT IS ANOTHER NAME FOR THE ACCUMULATOR
REGISTER?
*A-REGISTER
VERY GOOD.
THE A-REGISTER AND THE Q-REGISTER ARE USED ANYTIME THAT
THE CONTENTS OF A MEMORY LOCATION ARE TO BE MOVED FROM ONE
LOCATION TO ANOTHER AND ALSO THEY ARE USED TO DENOTE
OPERANDS FOR ARITHMETIC OPERATIONS. WHAT IS ANOTHER NAME
FOR THE MULTIPLIER-QUOTIENT REGISTER?
*Q-REGISTER
THAT'S RIGHT.
NOW IF THE PROBLEM WAS TO ADD (B) TO (C) AND STORE THE
RESULT IN D, THE FOLLOWING CODE SEGMENT WOULD BE REQUIRED,

```
  LDA B
  IAD C
  STA D
```

AFTER EXECUTION OF THIS CODE SEGMENT HAVE
THE (B) AND (C) BEEN CHANGED?
*NO THEY HAVE NOT
THAT'S RIGHT, ACCORDING TO THE DEFINITIONS OF LDA AND IAD
THE INITIAL CONTENTS OF THE OPERANDS ARE UNCHANGED.
WRITE A CODE SEGMENT WHICH WILL ADD (ADDEND) TO (A-REG).

*IAD ADDEND
VERY GOOD.
NOW WRITE A CODE SEGMENT TO COMPUTE; (SUM) = (FIRST) + (TP)
*LDA FIRST IAD TP STA SUM
THAT'S CORRECT. EITHER ONE OF FIRST OR TP COULD HAVE BEEN LOADED AND THE OTHER ADDED SINCE ADDITION IS COMMUTATIVE.
THE GENERAL FORM OF THE SUBTRACTION INSTRUCTION IS AS FOLLOWS;
**** ISB Y 15 INTEGER SUBTRACT *****
THIS INSTRUCTION
SUBTRACTS A COPY OF THE CONTENTS OF Y FROM THE CONTENTS OF THE A-REGISTER AND LEAVES THE DIFFERENCE IN THE A-REGISTER.
THE INITIAL CONTENTS OF THE A-REGISTER ARE DESTROYED; THE CONTENTS OF Y ARE UNCHANGED. GIVE THE MNEMONIC CODE FOR THE INSTRUCTION JUST DEFINED.
*ISB
THAT'S CORRECT.
WRITE AN INSTRUCTION SEGMENT WHICH WILL ADD (X) TO (Y) AND SUBTRACT (W) FROM THE SUM. STORE THIS RESULT IN A CELL NAMED RESULT, IN OTHER WORDS, COMPUTE RESULT = X+Y-W
*LDA Y IAD X ISB W STA RESULT
THAT'S CORRECT.
DO YOU WISH TO GO ON TO THE NEXT
SECTION?(TYPE ≠YES≠ OR ≠NO≠)
*NO
YOU HAVE COMPLETED LESSON ELATI
NOW TYPE ≠$LOGOFF≠ OR ≠$LESSON≠ TO EXECUTE
A NEW LESSON.
TYPE COMMAND
*$LESSON,ELATIC
THE UNCONDITIONAL JUMP INSTRUCTION HAS THE FOLLOWING GENERAL FORM;
**** UNJ Y 75 UNCONDITIONAL JUMP *****
THIS INSTRUCTION
TERMINATES THE CURRENT SEQUENCE OF CONTROL AND STARTS A NEW SEQUENCE AT LOCATION Y. THE NEXT INSTRUCTION EXECUTED IS (Y). THE CONTENTS OF ALL MEMORY LOCATIONS AND OF THE A AND Q REGISTERS ARE UNCHANGED. WRITE AN INSTRUCTION WHICH WILL CAUSE THE CURRENT SEQUENCE OF CONTROL TO BE TRANSFERRED TO A LOCATION NAMED RTJ.
*UNJ RTJ
THAT'S CORRECT.
INDEX REGISTERS IN ELASTIC MAY CONTAIN EITHER CONSTANTS OR ADDRESSES. NOTICE THAT IN THIS CONTEXT CONSTANTS MAY BE TREATED AS ADDRESSES AND VICE-VERSA. THINKING BACK, HOW MANY BITS ARE USED TO REPRESENT A NUMERIC ADDRESS IN THE MEMORY OF THE ELASTIC COMPUTER?
*12 BITS
VERY GOOD.
BO IS A SPECIAL INDEX REGISTER WHICH IS SET PERMANENTLY TO THE VALUE ZERO. UNLIKE THE OTHERS, ITS CONTENTS MAY NOT BE CHANGED AND IT MAY NOT BE USED FOR COUNTING. TYPE THE VALUE TO WHICH BO IS ALWAYS SET.
*ZERO

```
CORRECT.
AN INDEX REGISTER IS JUST LARGE ENOUGH TO CONTAIN AN
ADDRESS. BY MEANS OF THE INSTRUCTION ENX   A NUMBER MAY BE
PUT INTO AN INDEX REGISTER.  FOR EXAMPLE, IF WE WANTED TO
SET B6 TO A VALUE OF 5, THE INSTRUCTION TO PERFORM THIS
WOULD APPEAR AS;          ENX  5(6)  GIVE AN INSTRUCTION
WHICH WILL PLACE 23 IN INDEX REGISTER 3;
*INX 23(3)
ANSWER AGAIN.
*ENX 23(3)
THAT S GOOD.
UP TO THIS TIME ALL CONSTANTS IN ELASTIC WERE DECIMAL
CONSTANTS. IN ELASTIC 2  THE CAPABILITY IS ADDED WHICH
ALLOWS SPECIFICATION OF OCTAL CONSTANTS; THE CONSTANT IS
SUFFIXED BY THE CAPITAL LETTER =B=. FOR EXAMPLE, TO ENTER
46 (OCTAL) INTO INDEX REGISTER 4, ONE COULD CODE;
ENX 46B(4) REMEMBERING THAT 46 (OCTAL) = 38 (DECIMAL), THE
INSTRUCTION;          ENX 38(4) WOULD ALSO PLACE 46 OCTAL
INTO INDEX REGISTER 4 SINCE ALL NUMBERS ARE REPRESENTED
INTERNALLY AS THEIR BINARY EQUIVALENTS.  WRITE AN
INSTRUCTION WHICH WILL PLACE 100 (OCTAL) IN B7.
*ENX 100B(6)
TRY AGAIN PLEASE.
*ENX 100B(6)
ENTER ENX 100B(7) NOW.
*ENX 64(7)
THAT'S RIGHT, BUT IT WOULD HAVE BEEN EASIER TO CODE;
  ENX 100B(7)
TYPE COMMAND
*$LESSON,MACHEO
IT MAY BE HELPFUL AT THIS TIME TO ILLUSTRATE THE MACHINE
LANGUAGE NUMERIC EQUIVALENTS OF A FEW ELASTIC 2
INSTRUCTIONS SO AS TO FAMALIARIZE YOU WITH THE FORM
PRESENTED ABOVE. GIVEN THAT THE ADDRESS OF DATA IS 1234 AND
THE ADDRESS OF TEMP IS 7725 WE WILL NOW LIST SOME SYMBOLIC
INSTRUCTIONS AND THE CORRESPONDING MACHINE LANGUAGE
INSTRUCTIONS.
SYMBOLIC          MACHINE LANGUAGE (OCTAL)
ENX 2(7)          500700000002
ENX 35(3)         500300000043
ENX 35B(3)        500300000035
ENX DATA(6)       500600001234
ENX -1(5)         500500007776
ENX -TEMP(2)      500200000052
INX 2(7)          510700000002
INX -1(5)         510500007776
INX DATA(2)       510200001234
INX 24B(6)        510600000024
INX 52(3)         510300000064
LDX TEMP(1)       520100007725
LDX 12(3)         520300000014
LDX 1246B(7)      520700001246
STX DATA(5)       530500001234
```

```
STX 1437B(1)        530100001437
XHJ DATA(5),4       550400041234
XHJ DATA(3),27B     550300271234
XHF TEMP(2),46      550300567725
XHF 1246B(3),DATA 550312341246 (NOT GENERALLY DONE)
XEJ TEMP(1),25      540100317725
XEJ DATA(6),-3      540677741234
XEJ TEMP(7),-DATA 540765437725
```

NOW, STUDY THESE EXAMPLES UNTIL YOU FEEL THAT YOU COULD WRITE THE MACHINE LANGUAGE INSTRUCTION FOR ANY SYMBOLIC INSTRUCTION YOU MIGHT BE GIVEN. IF THE ADDRESS OF WORD IS 5246 WRITE THE MACHINE LANGUAGE INSTRUCTION FOR ENX WORD(3)

*500300005246

VERY GOOD, YOU SEEM TO UNDERSTAND.

IF THE ADDRESS OF CELL IS 1243 GIVE THE MACHINE LANGUAGE FOR LDX CELL(7) .

*520700001243

CORRECT.

IF THE ADDRESS OF WORD IS 1777 GIVE THE MACHINE LANGUAGE FOR XEJ WORD(2),17

*540200211777

VERY GOOD.

YOU MAY HAVE BEEN WONDERING WHY NOP AND ENX BOTH HAVE 50 AS THEIR NUMERIC OP CODE. THE ANSWER IS SIMPLE. SINCE THE SYMBOLIC INSTRUCTION; NOP RESULTS IN THE MACHINE INSTRUCTION; 500000000000 WHICH SAYS TO PUT A 0 IN BO, NOTHING HAPPENS. THIS IS DUE TO THE FACT THAT ANY INSTRUCTION WHICH ATTEMPTS TO CHANGE (BO) ACTS AS A DO-NOTHING STATEMENT BECAUSE (BO) CANNOT BE CHANGED. THAT IS, ENX 0(0) IS THE SAME THING AS NOP AND BOTH CAUSE NOTHING TO HAPPEN. HIT RETURN WHEN YOU HAVE READ THIS.

*

APPENDIX H

EVALUATION AND ATTITUDE QUESTIONNAIRES

On the following pages appear the various evaluation and attitude questionnaires that were used in the study. A brief explanation precedes each questionnaire.

The first questionnaire was used in the formative evaluation. The students in the lecture group answered questions 1 through 40 while the CAI group answered all of the questions. The original form of this questionnaire was first used by the College of Arts and Sciences at The University of Texas at Austin in 1968 in an evaluation of instructors teaching courses in the College.

C.S. 310 EVALUATION QUESTIONNAIRE

The University of Texas at Austin

This questionnaire is designed to help improve instruction, materials, and course structure. Through an honest, conscientious appraisal by you, the student, this evaluation questionnaire can be of great service in the continuing effort to improve the quality of the course.

Please rate items 1-33 according to the following rating scale. Place the numerical rating in the space provided on the separate answer sheet. Do not mark on this questionnaire. If the statement does not seem appropriate to the course, put no mark on the answer sheet for the item.

Very much in need of improvement				Average				Quite adequate
1	2	3	4	5	6	7	8	9

(On answer sheet, record rating of 1, 2, 3, 4, 5, 6, 7, 8, or 9.)

Lecture: PLEASE COMPLETE ITEMS 1 - 40

CAI: PLEASE COMPLETE ITEMS 1 - 58.

1. General benefit of this course to most individuals.

2. General approach of the instructor to the course and subject matter (i.e., how original, imaginative, interested in the course, and stimulating was the instructor?).

3. General helpfulness and value of the textbook used in the course.

4. Overall impression of the instructor when compared with your impression of an "ideal" instructor.

5. Fairness of the contents of tests in the course.

6. General benefit of the homework in the course.

7. Fairness of grading in the course.

8. Availability of equipment in labs for this course (such as computer facilities, supplies, etc.).

9. Actual realization of goals in this course when compared with goals you had set for the course.

10. Instructor's personal interest in students.

11. Daily preparation by the instructor (i.e., organization of the lecture or work to be carried out in class).

12. General benefit of lab in learning the information in this course.

13. Adequacy of the amount of required reading for this course in helping you learn the course information.

14. Clarity and understandability of tests in the course.

15. Challenge of tests in this course (i.e., how adequate was the challenge and stimulation provided by tests?).

16. Availability of the instructor for student consultation.

17. Instructor's demonstration of knowledge of the subject being taught.

18. Clarity of presentation of textbook in this course (including ease of understanding, examples, etc.).

19. Adequacy of amount of lab work (running programs on the computer) in learning the information for the course.

20. Adequacy of the amount of homework (written, etc.) in learning the information for the course.

21. Fairness of amount of weight placed on homework for grading purposes in the course.

22. Adequacy of the number of chances to bring up grades in this course.

23. The ability of material in this course (including reading, lectures, etc.) to hold or capture your interest.

24. Instructor's consideration and courtesy toward students in the course.

25. The probably overall rating of this course by the entire class.

26. Understandability of language used by the instructor.

27. Instructor's encouragement to student learning.

28. General organization of classroom procedures.

29. Rating of the instructor when compared with other college teachers you have had.

30. Adequacy of the teaching of the application of principles and concepts in this course.

31. Integration of homework with lectures (i.e., how well does the homework relate to classroom work?).

32. Instructor's freedom from annoying mannerisms.

33. Adequacy of the instructor's outlining of goals for the course, and how well he has developed these goals.

For items 34-40, indicate on the answer sheet the number of the one best choice.

34. Do all assignments require about the same length of time, or do some take much more time because of length or difficulty?

    (1) There is a great deal of variation
    (2) There is some variation
    (3) There is not much variation
    (4) There is almost no variation

35. The textbook is used mostly as:

    (1) background material for lectures (but the material is not repetitious of the lectures)
    (2) repetition of lecture material
    (3) test questions only (i.e., text material is not covered in any way whatsoever in lectures)
    (4) "busy-work" only

36. What grade do you think you will make in this course?

    (1) A
    (2) B
    (3) C
    (4) D
    (5) F

37. What grade do you think you deserve?

    (1) A
    (2) B
    (3) C
    (4) D
    (5) F

38. How would you classify this course?

    (1) formal and highly structured
    (2) partly formal and partly informal
    (3) informal
    (4) too informal and no structure at all.

39. Would you like to take another course dealing with this course's subject?

    (1) yes
    (2) no

40. If you have any comments or recommendations for improvement of this course, please indicate by writing your comments on the sheet labeled for this purpose in the answer sheet group.

Please rate items 41-53 according to the rating scale on page 1.

41. Overall impression of the computer-assisted instruction course that presented ELASTIC.

42. Adequacy of the CAI course as a suitable method for teaching ELASTIC.

43. Level of interest in ELASTIC fostered by the CAI course.

44. Rating of CAI course section dealing with number conversion review.

45. Rating of CAI course section dealing with instruction in ELASTIC 1.

46. Rating of CAI course section dealing with instruction in ELASTIC 2.

47. Rating of CAI course section dealing with instruction in ELASTIC 3.

48. Rating of CAI course section dealing with instruction in ELASTIC macros.

49. Rating of quizzes in CAI course.

50. Rating of facilities in CAI laboratory.

51. Rating of assistance given in CAI laboratory.

52. Rating of general environment in CAI laboratory.

For items 53-60, indicate on the answer sheet the number of the <u>one best choice</u>.

53. Was it better for you personally to participate in a self-paced learning situation as opposed to a formal classroom lecture environment?

    (1) much more beneficial
    (2) more beneficial
    (3) about the same
    (4) less beneficial
    (5) much less beneficial

54. If you had it to do over again, would you choose to take C.S. 310 through a CAI course?

    (1) yes
    (2) no

55. Had you had any previous exposure to the CAI method of learning prior to taking C.S. 310?

    (1) yes
    (2) no

56. Did you finish the program assignments before the due date?

    (1) long before
    (2) slightly before
    (3) on time
    (4) slightly after
    (5) long after

57. Did you have much trouble scheduling time to take instruction from the CAI course?

    (1) much trouble
    (2) a little trouble
    (3) no trouble.

58. Please express your opinions as to the advantages and disadvantages of computer-assisted instruction with respect to the course in ELASTIC as well as the usefulness of using CAI as a teaching method in general. Use the sheet labeled for this purpose in the answer sheet group.

EVALUATION QUESTIONNAIRE
ANSWER SHEET

Place a number expressing your rating for each of the questions on the evaluation sheet. (Answer questions 40 and 58 on the separate sheets provided for that purpose.)

1. ______	21. ______	41. ______
2. ______	22. ______	42. ______
3. ______	23. ______	43. ______
4. ______	24. ______	44. ______
5. ______	25. ______	45. ______
6. ______	26. ______	46. ______
7. ______	27. ______	47. ______
8. ______	28. ______	48. ______
9. ______	29. ______	49. ______
10. ______	30. ______	50. ______
11. ______	31. ______	51. ______
12. ______	32. ______	52. ______
13. ______	33. ______	53. ______
14. ______	34. ______	54. ______
15. ______	35. ______	55. ______
16. ______	36. ______	56. ______
17. ______	37. ______	57. ______
18. ______	38. ______	58. see attached sheet
19. ______	39. ______	
20. ______	40. see attached sheet	

----

40. This sheet is provided for comments to question 40 on the questionnaire.

----

58. This sheet is provided for comments on question 58 of the questionnaire.

The following instrument is the Semantic Differential used by the students to rate "AUTOMOBILE" and "COMPUTER."

COMPUTER

Please rate your attitude toward the word above according to the fifteen sets of bipolar adjectives below. For example, on set 1, if you think the word above is fast, circle one of the numbers between 7 and 4 to indicate the degree where 7 is the degree nearest fast. If you think the word above is slow, circle one of the numbers between 4 and 1 to indicate degree where 1 is the degree nearest slow.

****

fast	7	6	5	4	3	2	1	slow
interesting	7	6	5	4	3	2	1	dull
relaxed	7	6	5	4	3	2	1	tense
good	7	6	5	4	3	2	1	bad
fair	7	6	5	4	3	2	1	unfair
deep	7	6	5	4	3	2	1	shallow
valuable	7	6	5	4	3	2	1	worthless
active	7	6	5	4	3	2	1	passive
easy	7	6	5	4	3	2	1	difficult
flexible	7	6	5	4	3	2	1	inflexible
exciting	7	6	5	4	3	2	1	boring
pleasant	7	6	5	4	3	2	1	unpleasant
encouraging	7	6	5	4	3	2	1	depressing
safe	7	6	5	4	3	2	1	dangerous
strong	7	6	5	4	3	2	1	weak

****

Circle one choice in each statement below.

I. I was in the CAI / Lecture group.

II. I was in the Batch / TTY group.

The questionnaire that follows was given to the CAI group in the summative evaluation. The five responses for each question were assigned the numbers from 1 - 5 with the high score in each question assigned to the most positive response toward CAI. For some questions agreement with the statement indicated a positive response while for other disagreement indicates a positive response toward CAI. The direction in which each question was scored is written to the left of each statement number in the same questionnaire. The symbol "+" is used to denote a positive bias while "-" indicates negative bias. Included to the left of the "bias symbol" for each question is a number which indicates the value assigned to the leftmost response in each question.

This attitude questionnaire was originally developed by B. R. Brown and was reported in [6].

Questionnaire for CAI Students

Please circle the one choice which most nearly reflects your attitude relative to each statement presented below.

1 + 1. While taking Computer-Assisted Instruction I felt challenged to do my best work.

strongly disagree	disagree	uncertain	agree	strongly agree

5 - 2. I was concerned that I might not be understanding the material.

strongly disagree	disagree	uncertain	agree	strongly agree

5 - 3. I was not concerned when I missed a question because no one was watching me anyway.

strongly disagree	disagree	uncertain	agree	strongly agree

1 - 4. While taking Computer-Assisted Instruction I felt isolated and alone.

all the time	most of the time	some of the time	very seldom	never

1 - 5. I felt uncertain as to my performance in the programming course relative to the performance of others.

all the time	most of the time	some of the time	very seldom	never

1 - 6. I found myself just trying to get through the material rather than trying to learn.

all the time	most of the time	some of the time	very seldom	never

5 + 7. I knew whether my answer was correct or not before I was told.

all the time	most of the time	some of the time	very seldom	never

1 - 8. I guessed at the answers to questions.

all the time	most of the time	some of the time	very seldom	never

5 - 9. In a situation where I am trying to learn something, it is important to me to know where I stand relative to others.

strongly disagree	disagree	uncertain	agree	strongly agree

1 + 10. As a result of having studied some material by Computer-Assisted Instruction, I am interested in trying to find out more about the subject matter.

strongly disagree	disagree	uncertain	agree	strongly agree

1 - 11. I was more involved in running the machine than in understanding the material.

all the time	most of the time	some of the time	very seldom	never

1 + 12. I felt I could work at my own pace with Computer-Assisted Instruction.

strongly disagree	disagree	uncertain	agree	strongly agree

5 - 13. Computer-Assisted Instruction makes the learning too mechanical.

strongly disagree	disagree	uncertain	agree	strongly agree

1 + 14. I felt as if I had a private tutor while on Computer-Assisted Instruction.

strongly disagree	disagree	uncertain	agree	strongly agree

5 + 15. I was aware of efforts to suit the material specifically to me.

all the time	most of the time	some of the time	very seldom	never

1 - 16. I found it difficult to concentrate on the course material because of the hardware.

all the time	most of the time	some of the time	very seldom	never

1 - 17. Questions were asked which I felt were not relevant to the material presented.

all the time	most of the time	some of the time	very seldom	never

5 - 18. Computer-Assisted Instruction is an inefficient use of the student's time.

strongly disagree	disagree	uncertain	agree	strongly agree

1 - 19. While on a Computer-Assisted Instruction I encountered mechanical malfunctions.

all the time	most of the time	some of the time	very seldom	never

1 + 20. Computer-Assisted Instruction made is possible for me to learn quickly.

strongly disagree	disagree	uncertain	agree	strongly agree

5 - 21. I felt frustrated by the Computer-Assisted Instruction situation.

strongly disagree | disagree | uncertain | agree | strongly agree

5 - 22. The Computer-Assisted Instruction approach is inflexible.

strongly disagree | disagree | uncertain | agree | strongly agree

5 - 23. Even otherwise interesting material would be boring when presented by Computer-Assisted Instruction.

strongly disagree | disagree | uncertain | agree | strongly agree

1 + 24. In view of the effort I put into it, I was satisfied with what I learned while taking Computer-Assisted Instruction.

strongly disagree | disagree | uncertain | agree | strongly agree

1 + 25. In view of the amount I learned, I would say Computer-Assisted Instruction is superior to traditional instruction.

strongly disagree | disagree | uncertain | agree | strongly agree

1 + 26. With a course such as I took by Computer-Assisted Instruction I would prefer Computer-Assisted Instruction to traditional instruction.

strongly disagree | disagree | uncertain | agree | strongly agree

5 - 27. I am not in favor of Computer-Assisted Instruction because it is just another step toward depersonalized instruction.

strongly disagree | disagree | uncertain | agree | strongly agree

5 - 28. Computer-Assisted Instruction is too fast.

strongly disagree | disagree | uncertain | agree | strongly agree

5 - 29. Typing experience is necessary in order to perform easily on Computer-Assisted Instruction.

strongly disagree | disagree | uncertain | agree | strongly agree

5 - 30 Computer-Assisted Instruction is boring.

strongly disagree | disagree | uncertain | agree | strongly agree

---

Please circle one choise in the statement below.

I was in the Batch / TTY group.

The attitude questionnaire that follows was given to the subgroup of both groups that participated in the summative evaluation that used the teletype to submit their ELASTIC program assignments. This instrument was developed by the author.

Questionnaire for TTY Users

Please circle the one choice which most nearly reflects your attitude relative to each statement presented below.

1. Was it better for you personally to execute your program assignments in teletype mode as opposed to batch mode?
   (1) Much more beneficial
   (2) More beneficial
   (3) About the same
   (4) Less beneficial
   (5) Much less beneficial
2. Did you finish the program assignments before the due date?
   (1) Long before
   (2) Slightly before
   (3) On time
   (4) Slightly after
   (5) Long after
3. What was your experience scheduling time to use a teletype?
   (1) I had a lot of trouble
   (2) I had a little trouble
   (3) I had relatively no trouble
   (4) I had no trouble
4. How did the speed of the teletype affect your rate of progress?
   (1) Greatly hindered my progress
   (2) Hindered my progress
   (3) Had no effect on my progress
   (4) Moderately increased my progress
   (5) Greatly increased my progress
5. Did the noise of the teletype affect your concentration?
   (1) Had a great negative effect on my concentration
   (2) Had a small negative effect on my concentration
   (3) Had no effect on my concentration
   (4) Had a small positive effect on my concentration
   (5) Had a great positive effect on my concentration

6. How did having other people in the room affect you?
   (1) I found it to be very distracting
   (2) I found it to be distracting
   (3) It did not bother me
   (4) I found it to be helpful
   (5) I found it to be very helpful

7. How was your experience using the RESPOND time-sharing system?
   (1) RESPOND was frequently inoperative
   (2) I lost many files due to system malfunction
   (3) I lost a few files due to system malfunction
   (4) I had very little trouble with RESPOND
   (5) I had no trouble with RESPOND

8. What was your experience in learning to use RESPOND?
   (1) I had a great deal of trouble
   (2) I had some trouble
   (3) I had practically no trouble
   (4) I found it easy to learn

9. When did you use the teletype most?
   (1) Morning
   (2) Afternoon
   (3) Early evening
   (4) Night

10. How long was your average session at the teletype?
   (1) 1/2 hour
   (2) 1 hour
   (3) 1-1/2 hours
   (4) 2 hours
   (5) Over 2 hours

11. How many runs on the average did it take to debug your programs?
   (1) 1 to 3
   (2) 3 to 6
   (3) 6 to 10
   (4) 10 to 20
   (5) Over 20

12. To what degree did you use paper tape in conjunction with RESPOND?
    (1) A great deal
    (2) Some
    (3) A little
    (4) Not at all

13. To what degree did you punch cards from the teletype?
    (1) A great deal
    (2) Some
    (3) A little
    (4) Not at all

14. Was the amount of disk space available to you adequate for your purposes?
    (1) More than adequate
    (2) Adequate
    (3) Not enough
    (4) Not nearly enough

15. Please give a brief summary of your thoughts concerning the use of the teletype in a programming course such as C.S. 310.

Please circle one choice in the statement below.

I was in the CAI / Lecture group.

APPENDIX I

A TYPICAL PICLS AND RESPOND SESSION

The following pages constitute a typical session of interaction with the CAI course through PICLS and on-line execution of ELASTIC programs through RESPOND. Note that the interface between the CAI course and the ELASTIC assembler/interpreter is accomplished through using both conversational and nonconversational facilities of the RESPOND time-sharing system.

A TYPICAL PICLS AND RESPOND SESSION

THE UNDERLINED LINES DENOTE USER ENTERED LINES

∇DENOTES CONTROL AND BELL

```
CU.T.RESPOND .70
LOGIN 180DH««HJD CSAL0119
 TIME 11 53
 DATE 23 02 70
 PORT 15
.... SEE SKED FOR NEW RESPOND DUMP SCHEDULE
SHOW SKED
   STARTING FEB 23,1970. THE RESPOND SCHEDULE IS
    MONDAY-FRIDAY     0930 A.M. TO 0330 A.M.
    SATURDAY          0930 A.M. TO MIDNIGHT
  THE SCHEDULE FOR RESPOND DUMPS IS
  MONDAY-FRIDAY       0100 P.M. AN 0530 P.M. AN 330 A.M.
  SATURDAY            0100 P.M. AN 0530 P.M. AN MIDNITE
   THESE DUMPS CAUSE APPROX. A 25 MIN. DELAY, DURING
  WHICH RESPOND IS INACTIVE.  OTHER DUMPS WILL BE
  TAKEN IN THE EVENT OF A SYSTEM FAILURE.
  PLEASE NOTICE THAT STARTING TIMES ON ANY SCHEDULED
  PERIOD ARE APPROXIMATE START UP TIMES.  ACTUAL START UP
  TIME DEPENDS UPON THE LENGTH OF TIME REQUIRED FOR LOADING
  THE TRAY FOR RESPOND COMMON FILES WILL BE LOADED
  ONCE PER DAY, AT 0930.  USERS NOT GETTING THEIR
  JOBS INTO THAT TRAY MAY, OF COURSE, USE THE REGULAR
  INPUT TRAY FOR THE JOB.  IN THE LATTER CASE,
  THERE IS NO GUARANTEE THAT THE COMMON FILE WILL
  GET IN AT ANY PARTICULAR TIME.
.... FLST FILES
PRIVATE   FILES
PICLS  D      27 DIS
DISK SPACE ASSIGNED    40,USED       3
.... DELETE FILE PICLS D
....
```

CONVERSE PICLS INPUT=ELAS01 CAIDIR
SYSTEM JOB NAME X180YKX

```
*** PICLS READY FOR LOGON
TYPE USER NUMBER
*100
TYPE PASS WORD
*FHOM
TYPE COMMAND
*$LESSON,START
THE SECTIONS IN SUBCOURSE ELAST01 ARE
INTRO
STRUCT
PRELIM
HIT RETURN NOW
*
 DO YOU WISH TO GO ON TO THE NEXT
 SECTION (TYPE #YES# OR #NO#)
*YES
WELCOME TO THE COMPUTER-ASSISTED INSTRUCTION COURSE ON THE
PROGRAMMING LANGUAGE CALLED ELASTIC.  THIS COURSE IS
DESIGNED TO TEACH YOU THE FUNDAMENTAL CONCEPTS OF THE
ELASTIC LANGUAGE AND REQUIRES ABOUT 12 HOURS FOR
COMPLETION.  WHEN YOU HAVE COMPLETED THIS COURSE YOU
SHOULD BE ABLE TO WRITE COMPUTER PROGRAMS IN ELASTIC.
YOU SHOULD POSSESS A WORKING KNOWLEDGE OF THE BASIC
OPERATIONS OF THE RESPOND SYSTEM.  THIS MAY BE OBTAINED BY
READING THE PAPER BY BILL ALEXANDER WHICH IS AVAILABLE IN
COMPUTATION CENTER 1.  AN ADDITIONAL REFERENCE  THAT YOU
WILL NEED THROUGHOUT THE COURSE IS "THE ELASTIC
PROGRAMMING LANGUAGE REFERENCE MANUAL" BY FRED C. HOMEYER.
THROUGHOUT THE COURSE YOU WILL BE INTERACTING WITH THE
CDC 6600 COMPUTER SYSTEM THROUGH A COMPUTER PROGRAM CALLED
PICLS.  YOU WILL BE PRESENTED INFORMATION ON THE TELETYPE
AND THEN YOU WILL BE REQUIRED TO MAKE SOME RESPONSE.  TO
ENTER A RESPONSE TO A QUESTION, YOU MUST WAIT FOR AN * TO
APPEAR ON THE TELETYPE.  THE * IS A SIGNAL TO YOU THAT THE
PICLS SYSTEM IS AWAITING A RESPONSE FROM YOU.  AFTER
RECEIVING THE *, YOU SIMPLY TYPE IN YOUR ANSWER AND DEPRESS
THE RETURN KEY (LOCATED AT THE UPPER RIGHT HAND SIDE OF THE
KEYBOARD).  YOUR ANSWER WILL BE PROCESSED AND A MESSAGE
WILL BE RETURNED TO YOU REGARDING THE ACCURACY OF YOUR
ANSWER.  AFTER THIS OCCURS, FURTHER INSTRUCIONS OR
INFORMATION WILL BE PRESENTED.
TO CORRECT AN ERROR IN TYPING YOU HIT THE SHIFT KEY AND
THE LETTER O.  THE CHARACTER PRODUCED WILL ERASE ITSELF
AND THE CHARACTER JUST BEFORE IT.
NOW, JUST FOR PRACTICE, DEPRESS THE RETURN KEY.
*
```

```
$LESSON,PRELIM←←←←←←STRUCT
THE LISTING OF THE CAI COURSE STRUCTURE APPEARS BELOW.  YOU SHOULD KEEP THIS
LISTING FOR YOUR OWN REFERENCE AND USE IT AS A SORT OF MAP TO GUIDE YOU
THROUGH THE COURSE.  THE FORM USED IN THE LISTING WILL BE AS FOLLOWS,
"SUBCOURSE"
    "SECTION"
        DESCRIPTION OF CONTENTS OF SECTION NAMED ABOVE
    "SECTION"
        DESCRIPTION OF CONTENTS OF ABOVE SECTION
"SUBCOURSE"
    .
    .
    ETC.
PRESS RETURN TO RECEIVE THE LISTING.
*
      CAI COURSE STRUCTURE

ELAST01
    INTRO
        INTRODUCTION, HOW TO ENTER RESPONSES, ETC.
    STRUCT
        LIST OF SUBCOURSES AND SECTIONS
    PRELIM
        DESCRIPTION OF PREREQUISITES
ELAST02
    REVIEW
        BINARY NUMBERS
    REV1
        BINARY ARITHMETIC
    OCT
        BINARY AND OCTAL CONVERSIONS
ELAST03
    BEGIN
        COMMENTS ON ELASTIC COMPUTER
     SYMB
        DEFINITIONS; SYMBOL, CONSTANT, ETC.
    OPR
        CONCEPTS; OPERAND, OP CODE, ETC.
ELAST04
    ELAS1A
        CONCEPTS; ASSEMBLY LANGUAGE, CARD FORMAT, ETC.
    REGIS
        EXPLANATION OF A AND Q REGISTERS
ELAST05
    ELATI
        LDA, STA, IAD, ISB AND CODE SEGMENTS
    ELATIC
        UNJ, AZJ, AMJ, HLT, NOP AND CODE SEGMENTS
ELAST06
    IO
         RDI, PRI, RDO, PRO AND CODE SEGMENTS
    ELATIB
        IMU, IDV AND CODE SEGMENTS
    PSDO
        END, BSS, DEC, AND CODE SEGMENTS
```

```
ELAST07
    PROBI
        SAMPLE ELASTIC PROGRAM
    SAMP
        PROGRAM SEGMENTS, LOOPING, ADDRESS MODIFICATION
ELAST08
    QUIZI
        QUIZ OVER ELASTIC I
HIT RETURN PLEASE
*
ELAST09
    ELASII
        INTRODUCTION TO ELASTIC 2, CONCEPT OF B-BOX
     ADM
        BEA AND RELATED IDEAS
    PROBII
        LOOPING IN ELASTIC 2 WITH CODE SEGMENTS
ELAST10
    ELAIIB
        FORMAL DEFN. OF ENX, INX, LDX, STX, XEJ, XHJ
    MACHEQ
        MACHINE CODE EQUIVALENTS OF ELASTIC 2 INSTRUCTIONS
ELAST11
    ELAIII
        ARS, QRS, LRS AND CONCEPT OF SHIFTING
    LSHIFT
        ALS, QLS, LLS AND CODE SEGMENTS
ELAST12
    ENTER
        ENA, INA, ANA, ORA, COM
    MASK
        CODE SEGMENTS ON LOGICAL ARITHMETIC
    ALIO
        DISPLAY CODE, RDA, PRA AND CODE SEGMENTS
ELAST13
    LOG
        MASKING AND LOGICAL ARITHMETIC
    EXTPSD
        EXPRESSIONS IN ADDRESS FIELDS
    EXTIO
        EXTENDED I/O AND PSEUDO-INSTRUCTIONS
ELAST14
    QUIZII
        FIRST HALF OF QUIZ 2
    QUIZC
        SECOND HALF OF QUIZ 2
HIT RETURN PLEASE
*
ELAST15
    DIAGN
        EXPLANATION OF ERROR DIAGNOSTICS/ELASTIC
ELAST16
    MACROA
        INTRODUCTION TO MACROS
```

```
    MACAA
        FORMAL DEFNS. CONCERNING MACROS
ELAST17
    MACROB
        EXAMPLES OF MACROS AND IFT, IFF
    MACROC
        ERROR DIAGNOSTICS/MACROS
ELAST18
    MACROD
        CODE SEGMENTS WITH MACROS
HIT RETURN AND GO ONTO SECTION "PRELIM" FOR MORE MATERIAL.
*
 DO YOU WISH TO GO ON TO THE NEXT
 SECTION (TYPE #YES# OR #NO#)
*NO
 YOU HAVE COMPLETED*LESSON STRUCT
 NOW TYPE #$LOGOFF# OR #$LESSON# TO EXECUTE
 A NEW LESSON.
 TYPE COMMAND
*$ENDRUN
 LOGOFF AT 12.11.29 HOURS
 CENTRAL PROCESSOR TIME USED  38.879 SECONDS


 *** END OF PICLS RUN ***
JOB BACK ;
.... FLST FILES
PRIVATE   FILES
PICLS  D      26 DIS
DISK SPACE ASSIGNED    40,USED       3
.... DELE FILE PICLS D
.... CLEAR
.... ENTER ELAT
00000010=A       BSS 1               STORAGE AREA
00000020=B       BSS 1
00000030=ST      RDI A
00000040=        RDI B
00000050=        LDA A
00000060=        ISB B
00000070=        STA SUM
00000080=        PRI A
00000090=        PRI B
00000100=        PRI SUM
00000110=        PRO SUM
00000120=        HLT
00000130=SUM     BSS 1
00000140=        END ST
00000150=22
00000160=19
00000170=∇9=*ELAST1   EXAMPLE PROGRAM
00000170=∇EXIT
.... FILE PROG
```

```
.... CLEAR
.... SHOW PROG
*ELAST1     EXAMPLE PROGRAM
A       BSS 1                   STORAGE AREA
B       BSS 1
ST      RDI A
        RDI B
        LDA A
        ISB B
        STA SUM
        PRI A
        PRI B
        PRI SUM
        PRO SUM
        HLT
SUM     BSS 1
        END ST
22
19
.... SUBMIT ELAST INPUT=PROG OUTPUT=GOOD
SYSTEM JOB NAME     X180FLX
.... JOB BACK
```

```
FLST FILES
PRIVATE   FILES
PROG           17 DIS
ELAST  D        7 DIS
GOOD           45 DIS
DISK SPACE ASSIGNED   40,USED       9
.... DELETE FILE ELAST D
.... SHOW GOOD
1**************************************************************************
*******************************************************
0
0
0
          ELAST1     EXAMPLE PROGRAM

0
 BEGIN RUN

           0000   000000000000      A      BSS 1                STORA
GE AREA
           0001   000000000000      B      BSS 1
           0002   650000000000      ST     RDI A
           0003   650000000001             RDI B
           0004   120000000000             LDA A
           0005   150000000001             ISB B
           0006   200000000014             STA SUM
           0007   660000000000             PRI A
           0010   660000000001             PRI B
           0011   660000000014             PRI SUM
           0012   740000000014             PRO SUM
           0013   000000000000             HLT
           0014   000000000000      SUM    BSS 1
                                           END ST
OMULTIPLE-DEFINED SYMBOL. ADDRESS USED.

        *NONE*

        UNDEFINED SYMBOL. ADDRESS USED.

        *NONE*
1
 ASSEMBLY COMPLETE.
 NUMBER OF INSTRUCTIONS PROCESSED -    14
 STARTING LOCN 0002
 ASSEMBLY TIME -     .294 DECIMAL SECONDS
 (0000) = DECIMAL             22
 (0001) = DECIMAL             19
 (0014) = DECIMAL              3
 (0014) = OCTAL      000000000003
 HLT 000000000000 ENCOUNTERED AT LC = 0013. EXECUTION STOPPED.
 NUMBER OF INSTRUCTIONS EXECUTED -     10
 EXECUTION TIME -     .015 DECIMAL SECONDS
.... DELE FILE PROG FILE GOOD
....
CLEAR
....
```

CONVES+RSE PICLS INPUT=ELASO1 CAIDIR
SYSTEM JOB NAME X180GDX

*** PICLS READY FOR LOGON
TYPE USER NUMBER
*100
TYPE PASS WORD
*FHOM
TYPE COMMAND
*$LESSON,PRELIM
BEFORE BEGINNING TO LEARN ELASTIC YOU SHOULD BE FAMILIAR WITH THE BINARY AND OCTAL NUMBER SYSTEMS. IF YOU WOULD LIKE A REVIEW OF NUMBER SYSTEMS, THE NEXT SUBCOURSE YOU WILL WANT TO TAKE IS ELASTO2. OTHERWISE, YOU WILL WANT TO TAKE SUBCOURSE ELASTO3 NEXT.
BEING FAMILIAR WITH THE BASIC RESPOND SYSTEM COMMANDS WILL ALSO BE VERY HELPFUL TO YOU IN UTILIZING THE FULL CAPABILITIES AFFORDED BY THIS COURSE. INFORMATION ON RESPOND MAY BE OBTAINED BY READING BILL ALEXANDER'S PAPER WHICH IS AVAILABLE IN THE COMPUTATION CENTER. ALTHOUGH THE PAPER GIVES EXAMPLES IN FORTRAN, THE BASIC FILE MANIPULATION COMMANDS ARE JUST AS APPLICABLE WHEN DEALING WITH AN ELASTIC PROGRAM. YOU SHOULD NOW HIT THE RETURN KEY. WHEN ASKED IF YOU WANT TO GO TO THE NEXT SECTION YOU RESPOND WITH NO SINCE THIS IS THE END OF SUBCOURSE ELASTO1. TYPE SENDRUN WHEN REQUESTED TO TYPE LOGOFF OR LESSON. YOU SHOULD THEN RE-ENTER THE COURSE BY ENTERING A COMMAND OF FORM; CONVERSE PICLS INPUT=ELASTIJ WITH THE INPUT FILE AS SUBCOURSE ELASTO2 OR ELASTO3 DEPENDING UPON YOUR PREVIOUS BACKGROUND.
*
DO YOU WISH TO GO ON TO THE NEXT
SECTION (TYPE #YES# OR #NO#)
*NO
YOU HAVE COMPLETED LESSON PRELIM
NOW TYPE #$LOGOFF# OR #$LESSON# TO EXECUTE
A NEW LESSON.
TYPE COMMAND
*$ENDRUN
LOGOFF AT 12.25.05 HOURS
CENTRAL PROCESSOR TIME USED 28.837 SECONDS

*** END OF PICLS RUN ***
JOB BACK ;
.... CONVERSE PICLS INPUT=ELASO2 CAIDIR
SYSTEM JOB NAME X180GLX

*** PICLS READY FOR LOGON
TYPE USER NUMBER
*100
TYPE PASS WORD

```
*FHOM
 TYPE COMMAND
*$LESSON,START
THE SECTIONS IN SUBCOURSE ELASTO2 ARE
REVIEW
REV1
OCT
HIT RETURN NOW
*
 DO YOU WISH TO GO ON TO THE NEXT
 SECTION (TYPE #YES# OR #NO#)
*NO
 YOU HAVE COMPLETED LESSON START
 NOW TYPE #$LOGOFF# OR #$LESSON# TO EXECUTE
 A NEW LESSON.
 TYPE COMMAND
*$ENDRUN
 LOGOFF AT 12.28.33 HOURS
 CENTRAL PROCESSOR TIME USED  38.133 SECONDS


 *** END OF PICLS RUN ***
JOG BACK ;
.... FLST FILES
PRIVATE   FILES
PICLS  D      27 DIS
DISK SPACE ASSIGNED    40,USED       3
.... DELETE FILE PICLS D
.... LOGOUT
TOTAL*00.36.12
```

CONVERSE PICLS INPUT=ELAS14 CAIDIR
SYSTEM JOB NAME   X022YZX

```
 *** PICLS READY FOR LOGON
 TYPE USER NUMBER
*2
 TYPE PASS WORD
*FHOM
 TYPE COMMAND
*$LESSON,START
THE SECTIONS IN SUBCOURSE ELAST14 ARE
QUIZII
QUIZC
HIT RETURN NOW
*
 DO YOU WISH TO GO ON TO THE NEXT
 SECTION (TYPE #YES# OR #NO#)
*YES
YOU WILL NOW BE GIVEN A QUIZ
WHICH WILL TEST YOUR COMPREHENSION OF THE MATERIAL
PRESENTED. THIS QUIZ IS COMPOSED OF 60 QUESTIONS.  PLEASE
TYPE TEST IF YOU WANT TO TAKE THE QUIZ OR LOG OFF IF YOU
WANT TO LEAVE THE TEST FOR NEXT TIME.
*TEST
****************************************
1. WRITE AN INSTRUCTION WHICH WILL PLACE 4 IN INDEX
REGISTER 7.
*ENX 4(7)
2.   WRITE AN INSTRUCTION WHICH WILL INCREASE THE (B4) BY
13 OCTAL.
*INX 13B(4)
3.   WRITE AN INSTRUCTION WHICH WILL PLACE THE ADDRESS OF
SUM IN B2.
*ENX SUM(2)
4.   WRITE AN INSTRUCTION WHICH WILL PLACE THE LOWER 12
BITS OF (CELL) IN B6.
*LDX CELL(6)
5.   WRITE AN INSTRUCTION WHICH WILL STORE (B5) IN LOWER 12
 BITS OF A WORD NAMED HALT.
*STX HALT(5)
6. WRITE AN INSTRUCTION WHICH WILL JUMP TO OUT IF (B6)=3.
*XEJ OUT(6),3
7.   WRITE AN INSTRUCTION WHICH WILL JUMP TO EXIT IF (B3)
ARE GREATER THAN 14 DECIMAL.
*XHJ EXIT(3),14
8.   WRITE AN INSTRUCTION WHICH WILL ADD TO THE CURRENT
CONTENTS OF THE A-REGISTER, TO THE CONTENTS OF A WORD
WHOSE ADDRESS IS CONTAINED IN B4.
*IAD 0(4)
9.   WRITE AN INSTRUCTION WHICH WILL JUMP TO THE WORD WHOSE
 ADDRESS IS JP PLUS THE      CONTENTS OF B6.
```

*UNJ JP(6)
10. WRITE AN INSTRUCTION WHICH WILL LOAD THE A REGISTER WITH ITSELF. (USE CURRENT INSTRUCTION ADDRESS DESIGNATOR).
*LDA $
11. IF (B2) = 0014, EXPLAIN WHAT WILL OCCUR IF THE FOLLOWING INSTRUCTION IS GIVEN; XHJ JUMP(2),14
ANSWER BY TYPING JUMP OR NEXT SEQUENTIAL INSTRUCTION.
*NEXT SEQUENTIAL INSTRUCTION
12. IN THE INSTRUCTION STA CELL+6(3), WHAT IS THE PORTION BETWEEN "STA" AND "(3)" CALLED?
*IT IS CALLED THE BASE EXECUTION ADDRESS
ANSWER TRUE OR FALSE TO THE FOLLOWING QUESTIONS. TYPE T IF YOU THINK THE ANSWER IS TRUE AND F IF FALSE. 13. THE INSTRUCTION LDX 3426B(1), WHEN EXECUTED, CAUSES THE CONSTANT 3426B TO BE LOADED INTO INDEX REGISTER 1.
*F
14. THE INSTRUCTION INA TEMP WHEN EXECUTED, INCREASES THE A REGISTER BY THE ADDRESS OF TEMP.
*T
15. THE INSTRUCTION ENX 3777B(4), WHEN EXECUTED, CAUSES THE CONSTANT 3777B (NOT ALPHANUMERIC) TO BE ENTERED INTO B4.
*T
16. THE INSTRUCTION ENA 5, WHEN EXECUTED, CAUSES THE CONTENTS OF THE CELL WHOSE ADDRESS IS 0005 TO BE ENTERED INTO THE A REGISTER.
*F
17. IF THE INSTRUCTION COM IS GIVEN, THIS CAUSES THE SIGN BIT OF THE NUMBER IN THE A REGISTER TO BE REVERSED AND THE REMAINDER OF THE NUMBER IS UNCHANGED.
*F
18. IN THE ELASTIC COMPUTER, ALL LEFT SHIFT INSTRUCTIONS CAUSE THE CONTENTS OF THE REGISTER(S) INVOLVED TO BE SHIFTED LEFT END-OFF.
*F
19. THE INSTRUCTION XHF JUMP(2),14, WHEN EXECUTED, WILL CAUSE A TRANSFER TO JUMP IF (B2) = 0015.
*F
20. THE INSTRUCTION STX WORD(7), WHEN EXECUTED, CAUSES THE CONTENTS OF WORD TO BE STORED IN B7.
*F
21. WRITE AN INSTRUCTION WHICH WILL INCREASE THE (A REG) BY (B6).
*INA 0(6)
22. WRITE AN INSTRUCTION WHICH WILL PLACE THE DECIMAL CONSTANT 25 IN THE A REGISTER.
*ENA 25
23. WRITE AN INSTRUCTION WHICH WILL DECREASE (B4) BY 25 OCTAL.
*INX -25B(4)
24. WRITE AN INSTRUCTION WHICH WILL PLACE THE SIGN BIT (HIGH ORDER BIT) OF THE A REGISTER IN THE LOW ORDER BIT POSITION OF THE Q REGISTER.

*LRS 71
25. WRITE AN INSTRUCTION WHICH WILL SHIFT THE CONTENTS OF THE A REGISTER LEFT, END-AROUND 24 BITS.
*ALS 24
26. WRITE AN INSTRUCTION WHICH WILL SHIFT THE (Q REG) RIGHT END-OFF 7 BITS.
*QRS 7
27. CREATE A MASK NAMED MASK (USING CON) WHICH CAN BE USED TO EXTRACT THE LOWER 8 BITS OF A WORD.
*MASK CON 377B
28. WRITE A SEGMENT OF CODE USING (MASK) FROM QUESTION 27 WHICH WILL EXTRACT THE LOWER 8 BITS FROM A WORD NAMED TEMP AND HAVE THESE BITS IN THE A-REGISTER.
*LDA TEMP ANA MASK
29. GIVEN (WORD) = 000000001234 AND (CELL) = 123456774541 AND THE FOLLOWING CODE SEGMENT;
LDA WORD
ORA CELL
GIVE THE FINAL (A-REGISTER)
*123456775775
30. WRITE A SINGLE INSTRUCTION WHICH WILL READ DECIMAL VALUES FROM CARDS. STORE THE FIRST VALUE READ INTO A CELL NAMED X AND THE LAST VALUE INTO A WORD NAMED Z.
*RDI X,Z
YOU ARE NOW HALF FINISHED WITH THE QUIZ. TYPE SCORE.
*SCORE
97 PER CENT CORRECT OF 30 QUESTIONS
YOU SHOULD GO ON TO THE NEXT SECTION TO COMPLETE THE QUIZ. HIT RETURN NOW.
*
DO YOU WISH TO GO ON TO THE NEXT SECTION (TYPE #YES# OR #NO#)
*YES
31. WRITE A SINGLE INSTRUCTION WHICH WILL PRINT OCTAL VALUES. THE FIRST VALUE PRINTED SHOULD BE (Y+(B2)) AND THE LAST VALUE PRINTED SHOULD BE (X+(B6)).
*PRO Y(2),X(6)
32. WRITE AN INSTRUCTION WHICH WILL READ ALPHANUMERIC INFORMATION INTO CELLS NAMED A THROUGH B.
*RDA A,B
33. WRITE A SINGLE INSTRUCTION WHICH WILL CREATE DECIMAL CONSTANTS OF 22, 13, 44, 63, 1056 AND OCTAL CONSTANTS OF 23, 14, 72 AND 173.
*CON 22,13,44,63,1056,23B,14B,72B,173B
34. USING CON CREATE A HOLLERITH CONSTANT WHICH CONSISTS OF THE FOLLOWING MESSAGE; (NAME THE FIRST LOCATION WD)
TEXAS LONGHORNS
*WD CON HTEXAS,LONGHO,RNS
35. WRITE AN INSTRUCTION WHICH WILL PRINT ON A SINGLE LINE, THE MESSAGE IN QUESTION 34.
*PRA WD,WD+2
36. WRITE AN INSTRUCTION WHICH WILL JUMP TO A LOCATION FOUR ABOVE ITSELF.

*UNJ $-4
37. WRITE AN INSTRUCTION USING DEC WHICH WILL CREATE AN OCTAL CONSTANT DEPENDENT UPON THE ADDRESS ASSIGNED TO A LOCATION NAMED WORD.
*DEC WORD
38. GIVE THE INFORMATION REQUESTED FOR EACH OF THE FOLLOWING QUESTIONS, GIVEN THE ADDRESSES AND CONTENTS OF CELLS AND REGISTERS AS FOLLOWS; (EACH QUESTION IS SEPARATED FROM ALL OTHERS).

```
ADDRESS OF B = 0002        (B) = 000000000001
ADDRESS OF C = 0005        (C) = 777777777775
ADDRESS OF F = 0004        (F) = 000000000006
ADDRESS OF D = 0010        (D) = 000000000100
ADDRESS OF E = 0014        (E) = 000000000077
            (B1) = 0010          (B2) = 0005
            (B3) = 0010          (B4) = 7773
```

GIVEN THE INSTRUCTION LDA B+C(2) GIVE THE FINAL (A-REG).
*000000000077
39. GIVEN THE INSTRUCTION ENA B(3), GIVE THE FINAL CONTENTS OF THE A REGISTER.
*0000000000012←←←←←←←←←←←←←←000000000012
40. GIVEN THE INSTRUCTION ENX -3(2), GIVE THE FINAL CONTENTS OF B2.
*7774
41. GIVEN THE INSTRUCTION LDX C(2). GIVE THE FINAL (B2)
*7775
42. GIVEN THE INSTRUCTION INX -1(4), GIVE THE FINAL (B4).
*7772
43. GIVEN THE INSTRUCTIONS:
LDA C
ANA E
GIVE THE FINAL (A-REGISTER).
*000000000075
44. GIVEN THE CODE SEGMENT;
LDA D
ALS 3
STA D
GIVE THE FINAL (D).
*000000001000
45. GIVEN THE INSTRUCTION; ENX B(1). GIVE THE FINAL (B1).
*0002
46. GIVEN THE INSTRUCTION STX F+1(3), GIVE THE FINAL (C).
*777777770010
47. GIVEN THE CODE SEGMENT; ENA 77B LLS 6
AND THAT (Q-REG) = 000000000000 , GIVE THE FINAL CONTENTS OF THE A REGISTER.
*000000007700
48. GIVEN THE CODE; LDA D(4), GIVE THE FINAL (A-REGISTER).
*000000000006
49. GIVEN THE INSTRUCTION PRI C TYPE THE NUMBER THAT GETS PRINTED.
*-2

50. GIVEN THE INSTRUCTION PRO C, TYPE THE NUMBER THAT GETS PRINTED.
*777777777775
51. GIVEN THE CODE SEGMENT;
```
 LDA B
 INA 0(2)
```
GIVE THE FINAL (A-REGISTER).
*000000000006
52. GIVEN THE INSTRUCTION LDQ 2*B+F(4) , GIVE THE FINAL (Q REGISTER).
*000000000006
53. IF (WORD) = 000000000012, THE ADDRESS OF WORD IS 1777, AND THE INSTRUCTION WORD LDA $ IS GIVEN, GIVE THE FINAL (A REGISTER).
*120000001777
54. GIVEN THE CODE SEGMENT;
```
 LDA E
 LDQ Q(2)
 LLS 12
```
GIVE THE FINAL (Q-REGISTER).
*777777750000
55. GIVEN THE INSTRUCTION SEQUENCE;
```
 ENA 14
 ARS 36
```
GIVE THE FINAL (A-REGISTER).
*000000000000
56. WRITE AN INSTRUCTION SEGMENT WHICH WILL MULTIPLY THE CONTENTS OF A CELL WHOSE ADDRESS IS 0050 BY THE CONTENTS OF A WORD WHOSE ADDRESS IS CONTAINED IN B7.
*LDA 50B IMU 0(7)
57. WRITE AN INSTRUCTION WHICH WILL PLACE THE ADDRESS OF A WORD NAMED X IN B5.
*ENX X(5)
58. WRITE AN INSTRUCTION WHICH WILL ENTER THE ADDRESS OF ITSELF IN B3. (USE CURRENT INSTRUCTION ADDRESS).
*ENX $8←(3)
59. WRITE AN INSTRUCTION WHICH WILL SHIFT (AQ) LEFT THE NUMBER OF BIT POSITIONS SPECIFIED BY (B6).
*LLS 0(6)
60. WRITE AN INSTRUCTION SEGMENT WHICH WILL INCREASE THE PRESENT (A REGISTER) BY THE ADDRESS OF A WORD NAMED TP.
*INA TP
****
WOULD YOU LIKE TO SEE YOUR SCORE ON THIS QUIZ?
(ANSWER YES OR NO).
*YES
100 PER CENT CORRECT OF 30 QUESTIONS

NOW THAT YOU HAVE FINISHED YOUR QUIZ, WOULD YOU LIKE TO SEE THE CORRECT ANSWERS? ANSWER YES OR NO.

*YES

1. ENX 4(7)
2. INX 13B(4) OR INX 11(4)
3. ENX SUM(2)
4. LDX CELL(6)
5. STX HALT(5)
6. XEJ OUT(6),3
7. XHJ EXIT(3),14
8. IAD 0(4)
9. UNJ JP(6)
10. LDA $
11. NEXT SEQUENTIAL INSTRUCTION
12. BASE EXECUTION ADDRESS
13. F
14. T
15. T
16. F
17. F
18. F
19. F
20. F
21. INA 0(6)
22. ENA 25 OR ENA 31B
23. INX -25(4) OR INX -21(4)
24. LLS 1 OR LRS 71
25. ALS 24 OR ALS 30B
26. ORS 7
27. MASK CON 377B
28. LDA TEMP ANA MASK
29. 123456775775
30. RDI X,Z
31. PRO Y(2),X(6)
32. RDA A,B
33. CON 22,13,44,63,1056,23B,14B,72B,173B
34. WD CON HTEXAS ,LONGH,RNS
35. PRA WD,WD+2
36. UNJ $-4
37. DEC WORD
38. 000000000077
39. 000000000012
40. 7774
41. 7775
42. 7772
43. 000000000075
44. 000000001000
45. 0002
46. 777777770010
47. 000000007700
48. 000000000006
49. -2
50. 777777777775
51. 000000000006
52. 000000000006
53. 120000001777
54. 777777750000
55. 000000000000
56. LDA 50B IMU 0(7)
57. ENX X(5)
58. ENX $(3)
59. LLS 0(6)
60. INA TP

****

NOW THAT YOU HAVE THE CORRECT ANSWERS, YOU SHOULD GO BACK AND SEE WHAT YOUR MISTAKES WERE. PRESS RETURN WHEN YOU HAVE DONE THIS.

*_

```
 TYPE COMMAND
*$LESSON,MACROD
THIS COMPLETES THE DISCUSSION OF MACRO INSTRUCTIONS IN
ELASTIC. FOR BEST USE OF THE ABOVE DISCUSSION, YOU SHOULD
TRY WRITING SOME PROGRAMS WHICH USE MACROS.  REFERENCE
BACK TO THE MATERIAL DISCUSSED ABOVE SHOULD PROVE VALUABLE
IN YOUR EFFORTS.  JUST TO MAKE SURE THAT YOU UNDERSTAND
MACROS, LET'S CONSIDER SOME PROBLEMS.  ASSUME THAT YOU ARE
GIVEN THE FOLLOWING MACRO DEFINITION AND MACRO CALL;
      PLUS   MAC P1,P2,P3
             LDA P1
             IAD P2
             STA P3
      PLUS   END
      PLUS   MOP (ADD,ADD2,SUM)
TYPE THE INSTRUCTIONS GENERATED BY THE CALL.
*LDA ADD IAD ADD2 STA SUM
THAT'S CORRECT.
LET US TRY ANOTHER;
MACRO  MAC  P1=P2*P3/P4
       LDA P2
       IMU P3
       IDV P4
       STA P1
MACRO  END
MACRO  MOP (A,B,C,D)
GIVE THE INSTRUCTIONS IN THE MACRO
EXPANSION.
*LDA B IMU C IDV D STA A
YOU'RE RIGHT.
SUM   MAC P1,P2
      P1
      UNJ P2
SUB   END
SUB   MOP ((IMU CELL),JP)
GIVE THE INSTRUCTIONS GENERATED.
*IMU CELL UNJ JP
THAT'S VERY GOOD.
RT   MAC P1,P2,P3
     P1
     P2 SUM
     P3
RT   END
IF THE INSTRUCTIONS GENERATED FROM A CALL TO RT ARE;
     LDA CELL
     IAD SUM
     STA TOTAL
GIVE THE MACRO CALL.
*RT MOP ((LDA CELL),IAD,(STA TOTAL))
THAT'S CORRECT.
```

```
GIVE THE ONE CHARACTER ERROR CODE PRODUCED BY THE FOLLOWING
MACRO DEFINITION;
SUB   MAC P1,P2
      LDA P1
      STA P2
STOP  END
*T
NO, T WOULD BE USED TO INDICATE THE ABSENCE OF AN END
INSTRUCTION.  RESPOND AGAIN.
*O
TRY AGAIN PLEASE.
*L
THAT'S CORRECT.
GIVE THE ONE CHARACTER ERROR CODE PRODUCED BY THE FOLLOWING
MACRO DEFINITION;
SUB   MAC P1,P2
      LDA P1
      STA P2
*T
THAT'S RIGHT.
GIVE THE ONE CHARACTER ERROR CODE PRODUCED BY THE FOLLOWING
MACRO CALL;    SUBT  MOP (A,B,C,(IAD CELL)
*O
THAT IS RIGHT, NO CLOSING RIGHT PARENTHESIS WAS PRESENT.
```

REFERENCES

1. Adams, D. A. "PROCTOR: Program for the Control of Text Output and Student Responses," Report No. ATR-68(8112)-1, Aerospace Corporation, El Segundo, California, May, 1968.

2. Adams, Jeanne and Leonard Cohen. "Time-Sharing vs. Batch Processing: An Experiment In Programmer Training," Computers and Automation, March, 1969, 30 - 34.

3. Alpert, D. and D. L. Bitzer. "Advances in Computer-Based Education," Science, Vol. 167, No. 3925, March 20, 1970, 1582 - 1590.

4. Basic Computer Programming - Self-Instructional Course, Prospectus, Human Resources Research Office, George Washington University, 300 N. Washington Street, Alexandria, Va., March 8, 1968.

5. Beginning Basic, Dartmouth College, Hanover, New Hampshire, October, 1967.

6. Brown, B. R. "An Instrument for the Measure of Expressed Attitude Toward Computer-Assisted Instruction," Experimentation With Computer-Assisted Instruction in Technical Education, H. E. Mitzel and G. L. Bradon (eds.), (Semi-Annual Progress Report, Project No. OEC-5-85-074), University Park, Pa., The Pennsylvania State University, December 31, 1966.

7. Bryan, Glenn L. "Computers and Education," Computers and Automation, March, 1969, 16 - 19.

8. Bunderson, C. V. "The Computer and Instructional Design," Computer-Assisted Instruction, Testing and Guidance, W. E. Holtzman, editor, Harper and Row, Inc., 1970.

9. Computer-Assisted Instruction in Programming: AID Project, Progress Reports - Stanford Program In Computer-Assisted Instruction, July 1, 1968 to September 30, 1969.

10. Computer-Assisted Instruction In Programming: SIMPER and LOGO Project, Progress Reports - Stanford Program In Computer-Assisted Instruction, July 1, 1968 to September 30, 1969.

11. Conti, D. M. "Notes on DISPLAYTRAN," U.S. Naval Weapons Laboratory, Dahlgren, Va., March 23, 1970.

12. Control Data 6400/6500/6600 Computer Systems Reference Manual, Pub. No. 60100000, Control Data Corporation, 1968.

13. Feingold, S. L. "PLANIT- A Flexible Language for Computer-Human Interaction," AFIPS Conference Proceedings, Vol. 31, 1967 FJCC, 545 - 552.

14. Fenichel, R., Weizenbaum, J. and J. Yochelson. "A Program To Teach Programming," Communications of the ACM, Vol. 13, No. 3, March, 1970, 141 - 146.

15. FORTRAN Training - Outline of CAI Scheme, Brighton College of Technology, International Computers Ltd., England, April, 1969.

16. Homeyer, Fred C. A Macro Processor for the ELASTIC Programming Language, University of Texas Computation Center, TNN-78, 1968, Library of Congress Catalog No. QA76.H753.

17. Homeyer, Fred C. The ELASTIC Programming Language Reference Manual, A-Tex Printing Company, 1968, Austin, Texas.

18. Inservice Mathematics Education For Elementary School Teachers Via Computer-Assisted Instruction, Pennsylvania State University, Report No. R-25, November 21, 1969.

19. Kemery, John G. and John M. Nevison. "How to Write a TEACH Program at a Remote Teletype," Dartmouth College, Hanover, New Hampshire, December, 1967.

20. Kopstein, Felix and Robert J. Seidel. "Comments on Schurdak's 'An Approach to the Use of Computers in the Instructional Process'," American Educational Research Journal, Vol. 4, Number 4, November, 1967, 413 - 416.

21. Mathis, A., T. Smith and D. Hansen. "College Student's Attitudes Toward Computer-Assisted Instruction," Journal of Educational Psychology, 1970, Vol. 61, No. 1, 46 - 51.

22. Nevison, John M. "Suggestions For Teachers Using BASICT," Dartmouth College, Hanover, New Hampshire, September, 1968.

23. Osgood, C. E., Suci, G. J. and Tannenbaum. "The Measurement of Meaning," Urbana: University of Illinois Press, 1957.

24. Pearson, Erna, editor. RESPOND Users Manual, The University of Texas at Austin Computation Center, TRM 4.00, 1969.

25. PICLS User's Manual, Purdue University, March, 1969.

26. Rosenbaum, Feingold, Frye and Bennik. Computer-Based Instruction in Statistical Inference, Systems Development Corporation, Santa Monica, California, Technical Memorandum-2914/100/00, October 30, 1967.

27. Schurdak, John. "An Approach to the Use of Computers in the Instructional Process as an Evaluation," American Educational Research Journal, Vol. 4, Number 1, January, 1967, 59 - 73.

28. Schwartz, H. A. and H. S. Long. "Instruction By Computer," Datamation, September, 1966, 73 - 80.

29. SCOOP, International Computers and Tabulators, Ltd., London, England, March 12, 1968.

30. Seidel, Robert J. "Computers in Education: The Copernican Revolution In Education Systems," Computers and Automation, March, 1969, 24 - 29.

31. Seidel, Robert J. "Programmed Learning: Prologue to Instruction," Professional Paper 17-67, April 1967, Human Resources Research Office, George Washington University, 300 N. Washington Street, Alexandria, Va.

32. Seidel, R., Compton, J., Kopstein, F., Rosenblatt, R. and S. See. "Project IMPACT: Description of Learning and Prescription for Instruction," Professional Paper 22-69, June, 1969, Human Resources Research Office, George Washington University, 300 N. Washington Street, Alexandria, Va.

33. Seidel, Robert J. "Project IMPACT (Instructional Model Prototypes Attainable In Computerized Training)," Reports Control Symbol CSCRD-21(R1), December 2, 1966, Human Resources Research Office, George Washington University, 300 N. Washington Street, Alexandria, Va.

34. Seidel, Robert J. and H. G. Hunter. "The Application of Theoretical Factors in Teaching Problem Solving by Programmed Instruction," Technical Report, 1968, Human Resources Research Office, George Washington University, 300 N. Washington Street, Alexandria, Va.

35. Suppes, P., "The Uses of Computers in Education," Scientific American, September, 1966.